

**RANCANG BANGUN SISTEM PENJEJAKAN OBJEK  
MENGUNAKAN METODE VIOLA JONES  
UNTUK APLIKASI EYEBOT**

**Oleh :**

**Mukhlas Arihutomo**  
**NRP. 7207 030 060**

**Dosen Pembimbing :**

**Ir. Anang Budikarso, MT.**  
**NIP. 19630508 1988 03 1 003**

**Setiawardhana, ST.**  
**NIP. 19770824 2005 01 1 001**

**JURUSAN TEKNIK TELEKOMUNIKASI  
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
S U R A B A Y A  
JULI 2010**





## **PROYEK AKHIR**

# **RANCANG BANGUN SISTEM PENJEJAKAN OBJEK MENGUNAKAN METODE VIOLA JONES UNTUK APLIKASI EYEBOT**

**Oleh :**

**Mukhlas Arihutomo**  
**NRP. 7207 030 060**

**Dosen Pembimbing :**

**Ir. Anang Budikarso, MT.**  
**NIP. 19630508 1988 03 1 003**

**Setiawardhana, ST.**  
**NIP. 19770824 2005 01 1 001**

**JURUSAN TEKNIK TELEKOMUNIKASI  
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
S U R A B A Y A  
JULI 2010**



**RANCANG BANGUN SISTEM PENJEJAKAN OBJEK  
MENGUNAKAN METODE VIOLA JONES  
UNTUK APLIKASI EYEBOT**

**Oleh :**

**Mukhlas Arihutomo  
NRP. 7207.030.060**

**Proyek Akhir ini Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Ahli Madya (A, Md)  
di  
Politeknik Elektronika Negeri Surabaya  
Institut Teknologi Sepuluh Nopember Surabaya**

**Disetujui oleh :**

Tim Penguji Proyek Akhir :

Dosen Pembimbing :

**1.**  
**NIP.**

**Ir. Anang Budikarso, MT.**  
**NIP. 19630508 1988 03 1003**

**2.**  
**NIP.**

**Setiawardhana, ST.**  
**NIP. 19770824 2005 01 01**

**3.**  
**NIP.**

**Mengetahui :  
Ketua Jurusan Telekomunikasi**

\_\_\_\_\_  
**NIP.**



## ABSTRAK

Teknologi di bidang robotika sekarang ini telah berkembang dengan pesat dan sangat luas, terbukti dengan sudah diaplikasikannya robot pada banyak bidang. Robot juga dapat menggantikan pekerjaan yang tidak dapat dikerjakan oleh manusia. Adanya perkembangan teknologi tersebut menuntut munculnya suatu inovasi dari robot agar dapat lebih bermanfaat, salah satunya sebagai sistem keamanan.

Pada Proyek Akhir ini, telah dibuat sebuah robot yang terintegrasi bernama Eyebot. Eyebot adalah sebuah robot yang memiliki kemampuan mencari, mendeteksi, dan menangkap adanya gerakan dari suatu objek. Eyebot merupakan sebuah aplikasi dari ilmu *Computer Vision* dengan menggunakan metode yang ada dalam bidang *Image Processing*. Eyebot dilengkapi sebuah kamera *webcam* yang berfungsi sebagai mata yang memungkinkan robot mampu mendeteksi dan menangkap adanya gerakan dari suatu objek. Selain itu, eyebot dilengkapi dengan dua buah motor DC yang dapat berputar dengan ke arah kanan-kiri dan ke arah atas-bawah untuk menggerakkan kamera tetap mengarah ke target. PC sebagai otak eyebot bertugas mengolah citra yang masuk, mengambil keputusan, serta memerintahkan ke rangkaian kontrol untuk menggerakkan bagian-bagian robot sesuai dengan kondisi target. Diharapkan Eyebot dapat bermanfaat sebagai sebuah sistem keamanan dengan tingkat ketelitian yang tinggi.

**Kata Kunci :** *Eyebot, robotika, computer vision, image processing, webcam, DC motor.*





## ABSTRACT

Technology in robotics field today has grown rapidly and very spacious, it has been proved by robots are used in the many sectors. Robots also can replace the jobs that can not be done by humans. The development of these technologies demand the appearance of an innovation of robot to be more beneficial, one of them as a security system.

In this final project, I will make integrated robot named Eyebot. Eyebot is a robot that has ability to find, detect and capture the movement of an object. Eyebot is an application of Computer Vision science using the existing Image Processing methods. Eyebot is equipped with a camera webcam as its eye that can detect and capture the movement of an object. Beside that, eyebot is equipped with two DC motors that can rotate to right-left and top-down direction to move the camera towards the target. PC as robot's brain process the incoming image, make decisions, and give instruction to the control circuit to move the robot parts accordance with target condition. I hope that Eyebot can be useful as a security system with a high level of accuracy.

**Keyword :** *Eyebot, robotics, computer vision, image processing, webcam, DC motor.*



## KATA PENGANTAR



Syukur Alhamdulillah kami panjatkan kepada Allah SWT karena hanya dengan rahmat, hidayah dan inayah-Nya kami dapat menyelesaikan proyek akhir ini dengan judul :

### **RANCANG BANGUN SISTEM PENJEJAKAN OBJEK MENGUNAKAN METODE VIOLA JONES UNTUK APLIKASI EYEBOT**

Dalam menyelesaikan proyek akhir ini, kami berpegang pada teori yang pernah kami dapatkan dan bimbingan dari dosen pembimbing proyek akhir. Dan pihak – pihak lain yang sangat membantu hingga samapi terselesaikannya proyek akhir ini.

Proyek akhir ini merupakan salah satu syarat akademis untuk memperoleh gelar Ahli Madya (A.Md.) di Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh Nopember Surabaya.

Kami menyadari bahwa masih banyak kekurangan pada perancangan dan pembuatan buku proyek akhir ini. Oleh karena itu, besar harapan kami untuk menerima saran dan kritik dari para pembaca. Semoga buku ini dapat memberikan manfaaat bagi para mahasiswa Politeknik Elektronika Negeri Surabaya pada umumnya dan dapat memberikan nilai lebih untuk para pembaca pada khususnya.

Surabaya, Juli 2010

Penyusun



## UCAPAN TERIMA KASIH

Puji syukur yang tak terhingga saya sampaikan kepada Allah SWT Yang Maha Berkuasa Atas Segalanya, karena hanya dengan ridho, hidayah dan anugerah-Nya saya dapat menyelesaikan proyek akhir ini. Ucapan terima kasih saya sampaikan juga ke berbagai pihak yang turut membantu memperlancar penyelesaian proyek akhir ini, yaitu kepada:

- ☺ Kedua orang tuaku Bapak, Ibu yang banyak memberikan Doa, Kasih Sayang, Cinta, Kesabaran sejak aku dalam kandungan serta bimbingan dan semangat sampai aku menjadi sekarang ini, terima kasih banyak atas SEMUA yang telah Bapak dan Ibu berikan.
- ☺ Mbak Ana yang aku sayangi.
- ☺ Bapak **Ir. Dadet Pramadihanto, M.Eng, Ph.D**, selaku Direktur Politeknik Elektronika Negeri Surabaya.
- ☺ Bapak **Arifin, ST, MT**, selaku Ketua Jurusan Teknik Telekomunikasi Politeknik Elektronika Negeri Surabaya.
- ☺ Bapak **Ir. Anang Budikarso, MT**. dan bapak **Setiawardhana, ST** selaku dosen pembimbing yang telah sabar dan arif dalam membimbing dan mendampingi saya serta banyak membantu selama pengerjaan Proyek Akhir ini. Mohon maaf bila ada tindakan maupun perkataan saya yang kurang berkenan di hati bapak dan terima kasih banyak atas saran, nasehat, dan ilmu yang diberikan kepada saya, semoga bermanfaat dimasa yang akan datang. Amin.
- ☺ **Temen-temen D3TB '07** terima kasih banyak sudah terima aku di lingkungan kalian. I will always miss u all...
- ☺ Semua pihak yang belum saya sebutkan, yang telah membantu saya baik selama perkuliahan maupun dalam pengerjaan proyek akhir ini. Terima kasih.

Segala ucapan terima kasih tentunya belum cukup, semoga Allah SWT senantiasa membalas segala kebaikan kalian. Amin.



# DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>iv</b>
<b>ABSTRACT .....</b>	<b>v</b>
<b>KATA PENGANTAR .....</b>	<b>vi</b>
<b>UCAPAN TERIMA KASIH .....</b>	<b>vii</b>
<b>DAFTAR ISI .....</b>	<b>viii</b>
<b>DAFTAR GAMBAR .....</b>	<b>x</b>
<b>DAFTAR TABEL .....</b>	<b>xii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	1
1.3 Batasan Masalah.....	2
1.4 Tujuan .....	2
1.5 Metodologi .....	2
1.6 Sistematika Pembahasan .....	5
<b>BAB II PERANCANGAN SISTEM DAN TEORI PENUNJANG .....</b>	<b>7</b>
2.1 Perancangan Perangkat Lunak .....	7
2.1.1 Program Pada Mikrkontroler.....	7
2.1.2 Program Pada PC (Komputer).....	9
2.1.2.1 OpenCV Library.....	10
2.1.2.2 Metode Viola-Jones.....	12
2.2 Perancangan Perangkat Keras .....	14
2.2.1 Kamera .....	16
2.2.2 Personal Computer (PC).....	16
2.2.3 Minimum System .....	17
2.2.3.1 Kontroler ATmega 8535 .....	17
2.2.3.1.1 Konfigurasi Pin ATmega 8535 .....	18
2.2.3.1.2 Karakteristik ATmega 8535 .....	19
2.2.3.2 Metode Komunikasi .....	20
2.2.3.2.1 Komunikasi Antara Kamera Dengan PC .....	20
2.2.3.2.2 Komunikasi Antara PC Dengan Kontroller.....	21
2.2.4 Motor DC .....	22

<b>BAB III PEMBUATAN SISTEM, PENGUJIAN DAN ANALISA</b>	<b>25</b>
3.1 Pembuatan Sistem.....	25
3.1.1 Pembuatan Perangkat Keras .....	25
3.1.1.1 Minimum System.....	25
3.1.1.2 Konstruksi Sistem .....	29
3.1.2 Pembuatan Software .....	32
3.1.2.1 Mengaktifkan Kamera.....	31
3.1.2.2 Deteksi Wajah.....	31
3.1.2.3 Perhitungan Posisi.....	32
3.1.2.4 Transmisi Data Serial.....	33
3.1.2.5 Program Kontroler .....	36
3.2 Pengujian dan Analisa.....	40
3.2.1 Pengujian dan Analisa Perangkat Keras .....	40
3.2.1.1 Pengujian Mikrokontroler ATmega 8535.....	40
3.2.1.2 Pengujian dan Analisa Modul LCD .....	41
3.2.2.3 Pengujian dan Analisa Rangkaian Driver .....	42
3.2.2.4 Pengujian dan Analisa Komunikasi Serial .....	45
3.2.2 Analisa Perangkat Lunak .....	46
3.2.2.1 Analisa Program Pada PC .....	46
3.2.2.2 Analisa Program Pada Kontroler .....	50
3.2.3 Pengujian Sistem Secara Keseluruhan .....	52
<b>BAB IV PENUTUP</b>	<b>55</b>
4.1 Kesimpulan .....	55
4.2 Saran .....	55
<b>DAFTAR PUSTAKA.....</b>	<b>56</b>



## DAFTAR GAMBAR

<b>Gambar 1.1</b>	Kontroler ATmega 8535.....	4
<b>Gambar 1.2</b>	Desain Mekanik EyeBot .....	4
<b>Gambar 2.1</b>	Icon Code Vision AVR.....	7
<b>Gambar 2.2</b>	Tampilan Awal Code Vision AVR .....	8
<b>Gambar 2.3</b>	Konfigurasi chip pada Code Wizard AVR.....	8
<b>Gambar 2.4</b>	<i>Flowchart</i> program .....	9
<b>Gambar 2.5</b>	Icon OpenCV Installer .....	11
<b>Gambar 2.6</b>	Icon Aplikasi MS Visual Studio .....	11
<b>Gambar 2.7</b>	Konfigurasi OpenCV pada MS Visual Studio .....	11
<b>Gambar 2.8</b>	Contoh fitur yang digunakan Viola-Jones.....	12
<b>Gambar 2.9</b>	<i>Integral Image</i> .....	13
<b>Gambar 2.10</b>	<i>Cascade Classifier</i> .....	14
<b>Gambar 2.11</b>	Diagram EyeBot .....	15
<b>Gambar 2.12</b>	<i>Webcam</i> .....	16
<b>Gambar 2.13</b>	Laptop sebagai pengganti PC.....	16
<b>Gambar 2.14</b>	Diagram minimum system ATmega 8535.....	17
<b>Gambar 2.15</b>	Konfigurasi <i>pin out</i> Atmega 8535.....	18
<b>Gambar 2.16</b>	Grafik respon arus dan frekuensi .....	20
<b>Gambar 2.17</b>	Jenis Konektor USB .....	21
<b>Gambar 2.18</b>	Penetapan kaki konektor .....	21
<b>Gambar 2.19</b>	Port DB9 Jantan .....	22
<b>Gambar 2.20</b>	Topologi H-Bridge.....	23
<b>Gambar 2.21</b>	Motor DC dan <i>Gearbox</i> .....	23
<b>Gambar 3.1</b>	Diagram dan rangkaian <i>Minimum System</i> . .....	25
<b>Gambar 3.2</b>	<i>Schematic</i> rangkaian <i>supply</i> . .....	26
<b>Gambar 3.3</b>	<i>Schematic</i> rangkaian serial.....	26
<b>Gambar 3.4</b>	<i>Schematic</i> rangkaian ATmega 8535.....	27
<b>Gambar 3.5</b>	<i>Schematic</i> driver motor .....	28
<b>Gambar 3.6</b>	Modul LCD 16x2 .....	28
<b>Gambar 3.7</b>	Konstruksi Robot .....	29
<b>Gambar 3.8</b>	Tempat pemasangan 2 buah motor .....	30
<b>Gambar 3.9</b>	Hasil <i>capture</i> dari kamera.....	31
<b>Gambar 3.10</b>	Hasil deteksi wajah .....	32
<b>Gambar 3.11</b>	Led menyala.....	40
<b>Gambar 3.12</b>	Berhasil menampilkan tulisan pada LCD .....	41
<b>Gambar 3.13</b>	<i>Schematic</i> driver motor .....	42

<b>Gambar 3.14</b>	Komponen IC driver motor pada rangkaian .....	42
<b>Gambar 3.15</b>	Ilustrasi driver motor .....	43
<b>Gambar 3.16</b>	Transistor yang aktif dan arah arus listrik .....	44
<b>Gambar 3.17</b>	IC Serial pada rangkaian.....	45
<b>Gambar 3.18</b>	Menampilkan tulisan melalui serial.....	46
<b>Gambar 3.19</b>	Simulasi tampilan frame .....	47
<b>Gambar 3.20</b>	Pengujian 1 .....	48
<b>Gambar 3.21</b>	Pengujian 2 .....	48
<b>Gambar 3.22</b>	Pengujian 3 .....	49
<b>Gambar 3.23</b>	Pengujian 4 .....	49
<b>Gambar 3.24</b>	Aplikasi konsol.....	50

## DAFTAR TABEL

<b>Tabel 2.1</b>	Karakretistik Elektrik .....	19
<b>Tabel 2.2</b>	Jenis Sinyal RS23 .....	22
<b>Tabel 3.1</b>	Tabel kebenaran aksi robot.....	33
<b>Tabel 3.2</b>	Tabel kebenaran arah gerak motor .....	43
<b>Tabel 3.3</b>	Pengujian deteksi wajah berdasar jarak.....	52
<b>Tabel 3.4</b>	Pengujian kecepatan respon sistem .....	53



# **BAB I**

## **PENDAHULUAN**



# BAB 1

## PENDAHULUAN

### 1.1. LATAR BELAKANG

Perkembangan teknologi dibidang sistem keamanan sangat diperlukan. Hingga saat ini, beberapa jenis sistem pengaman sudah diterapkan dibanyak tempat. Tetapi kebanyakan sistem yang dipakai bersifat pasif, karena hanya memberikan peringatan. Seperti halnya sistem alarm, sistem autentikasi pin, pintu barcode, ataupun lemari brankas. Dengan hanya memberikan aksi-aksi pasif tersebut, sistem pengaman dapat dengan mudah dianalisa dan ditembus tanpa meninggalkan jejak si pencuri atau para pelaku. Untuk itu diperlukan sistem pengaman yang dapat menyimpan aktifitas orang yang dilihatnya. Dengan harapan, itu semua bisa menjadi barang bukti bahwa ada orang tidak dikenal yang pernah masuk rumah tanpa ijin, sehingga pemilik dapat menentukan tindakan selanjutnya.

Dalam Proyek Akhir ini kami akan membuat sebuah sistem pengaman yang memanfaatkan ilmu robotika dari bidang *Computer Vision* yang memakai metode image processing. Dengan menggabungkan kedua ilmu tersebut, akan dibangun sebuah robot pengawas yang mampu melakukan scanning pada sebuah ruangan. Robot ini dilengkapi dengan sebuah kamera yang selalu aktif mengolah citra yang didapat sehingga dapat mendeteksi adanya gerakan dari sebuah objek tertentu. Selain itu, sistem ini dapat digunakan sebagai alat yang dapat mengenali dan mencari benda tertentu karena ketelitian tinggi yang dimiliki oleh sistem.

### 1.2. PERUMUSAN MASALAH

Adapun permasalahan yang dihadapi dalam penyelesaian proyek akhir ini adalah sebagai berikut :

- Bagaimana cara robot mendeteksi objek wajah?
- Bagaimana cara menentukan target dari banyak objek wajah yang terdeteksi?
- Bagaimana cara mengatur pergerakan kamera agar tetap memantau objek wajah yang terdeteksi?
- Bagaimana cara melakukan komunikasi antara kontroler dengan PC?

### 1.3. BATASAN MASALAH

Di dalam permasalahan tersebut akan diberi batasan-batasan masalah sebagai berikut :

- Deteksi objek wajah menggunakan sebuah *webcam* yang dikombinasikan dengan program pada PC.
- Objek yang dideteksi berupa wajah yang terlihat utuh dari depan dengan jarak dari robot  $< 5$  meter.
- Aktuator utama berupa 2 buah motor DC yang dikendalikan oleh mikrokontroler.
- PC memberikan perintah kepada kontroler melalui komunikasi serial.

### 1.4. TUJUAN

Tujuan dari proyek akhir ini adalah membuat sebuah robot yang dapat mendeteksi adanya objek manusia yang memakai parameter wajah sebagai acuannya dan juga memiliki beberapa kemampuan sederhana yang dapat diaplikasikannya sebagai sistem keamanan.

### 1.5. METODOLOGI

Metodologi yang digunakan dalam proyek akhir ini meliputi:

#### 1.5.1 Studi Literatur

Pada tahap ini, dilakukan studi literatur mengenai metode-metode serta referensi program dan source code yg dibutuhkan dalam menyelesaikan proyek ini.

#### 1.5.2 Perancangan Sistem

##### 1.5.2.1 Perancangan Perangkat Lunak

Pada tahap perancangan sistem perangkat lunak ini, dilakukan beberapa percobaan untuk menentukan metode mana yang lebih baik dan lebih sesuai untuk diaplikasikan pada sistem ini. Pada sistem ini terdapat 2 bagian program yaitu :

1. Program pada *microcontroller*
2. Program pada PC

Program pada *microcontroller* ini akan didisain dengan tujuan agar dapat menerima perintah dari PC untuk menggerakkan aktuator sesuai kondisi objek yang ada.

Sedangkan Program untuk PC, didesain agar PC dapat mengambil gambar dari kamera dan mengolah gambar tersebut untuk diambil informasinya dari objek wajah yang terdeteksi.



Setelah mendapatkan informasi, diharapkan program dapat mengambil keputusan langkah apa yang berikutnya akan diambil.

### **1.5.2.2 Perancangan Perangkat Keras**

Dalam tahap perancangan perangkat keras ini, akan dilakukan perancangan fisik dari robot dan perancangan PCB dari rangkaian. Untuk perancangan PCB, akan dibuat sebuah rangkaian yang memiliki fitur-fitur yang diperlukan dalam menjalankan sistem ini.

Sedangkan untuk perancangan fisik robot, akan ilustrasikan beberapa desain robot. Desain dari robot ini lebih menyerupai robot kepala dengan satu mata. Robot ini harus bisa mengarahkan kamera berikut senjatanya ke kanan-kiri dan ke atas-bawah. Robot ini adalah robot yang tidak berpindah tempat, dan sebisa mungkin didesain untuk bisa ditempatkan dimana saja. Selain disain, harus juga dipertimbangkan bahan utama penyusun robot. Bahan tersebut haruslah bahan yang mudah dibentuk, ringan tetapi kuat.

### **1.5.3 Pembuatan Sistem**

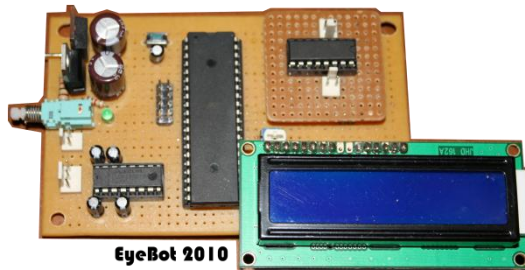
#### **1.5.3.1 Pembuatan Perangkat Lunak**

Pembuatan proyek akhir ini menggunakan Visual Studio 6.0 dengan bahasa C yang diintegrasikan dengan library OpenCV. OpenCV atau Open Computer Vision sendiri adalah sebuah library yang dengan sengaja dirancang untuk membantu dalam proyek-proyek *computer vision*.

Tujuan dari penggunaan OpenCV adalah untuk mempermudah dalam penggunaan metode-metode yang diperlukan. Hal tersebut dikarenakan, pada library OpenCV terdapat fungsi-fungsi tentang *image processing* yang dapat dipanggil apabila diperlukan.

#### **1.5.3.2 Pembuatan Perangkat Keras**

Untuk pembuatan perangkat keras, dilakukan pada desain mekanik robot dan rangkaian mikrokontroler yang telah dimodifikasi yang terdiri dari R. Supply, R. Serial, R. Motor Driver, R. LCD, dan R. MinSys ATmega 8535.



**Gambar 1.1** Kontroler ATMega 8535

Sedangkan ilustrasi desain mekanik EyeBot dapat dilihat pada gambar di bawah ini :



**Gambar 1.2** Desain mekanik EyeBot

#### **1.5.4 Integrasi Perangkat Lunak dan Perangkat Keras**

Setelah perangkat keras dan perangkat lunak telah selesai dibuat, langkah berikutnya adalah mengintegrasikan antara keduanya. Sarana yang digunakan komunikasi serial.

Dengan komunikasi tersebut, PC atau komputer utama dapat memberikan perintah kepada perangkat luar untuk melakukan aksi yang sesuai dengan keinginan kita.

#### **1.5.5 Pengujian dan Analisa Sistem**

Pengujian system dari robot dilakukan pada beberapa aspek, antara lain; kecepatan deteksi objek, kecepatan dan keakuratan robot dalam mengejar objek, dan waktu yang dibutuhkan untuk meng-*capture* target. Diharapkan setelah robot selesai dibuat, memiliki output yang mendekati sempurna, yaitu deteksi objek yang cepat, selalu memantau target kemanapun dia berpindah tempat dalam jangkauan tertentu, dan dapat meng-*capture* target.

#### **1.5.6 Pembuatan Laporan**

Pembuatan laporan proyek akhir dan melakukan publikasi pada seminar ilmiah. Serta membuat dokumentasi dari semua tahapan proses diatas berupa laporan yang berisi tentang dasar teori, hasil proyek akhir, serta hasil analisa.

### **1.6 SISTEMATIKA PEMBAHASAN**

Buku proyek akhir ini terdiri dari 5 (lima) bab, dimana masing-masing bab mempunyai kaitan satu sama lain, bab-bab yang akan dibahas adalah sebagai berikut :

#### **BAB I PENDAHULUAN**

Bab ini berisi tentang latar belakang, tujuan, perumusan masalah, batasan masalah, metodologi dan sistematika pembahasan proyek akhir ini.

#### **BAB II PERANCANGAN SISTEM DAN TEORI PENUNJANG**

Bab ini berisi tentang perencanaan sistem yang dibuat dan dasar teori untuk menunjang penyelesaian masalah dalam proyek akhir ini. Teori dasar yang diberikan meliputi: *OpenCV*, *Image Processing*, *Minsys* *ATMega* dll.

**BAB III PEMBUATAN SISTEM, PENGUJIAN DAN ANALISA**

Bab ini berisi tentang pembuatan sistem baik perangkat lunak maupun perangkat kerasnya, pengukuran dan analisa dari hasil sistem yang sudah dibuat..

**BAB IV PENUTUP**

Bab ini berisi kesimpulan dan saran serta rencana pengembangan proyek akhir, jika dimungkinkan untuk masa yang akan datang

**DAFTAR PUSTAKA**

Pada bagian ini berisi tentang referensi – referensi yang telah dipakai oleh penulis sebagai acuan dan penunjang serta parameter yang mendukung penyelesaian proyek akhir ini baik secara praktis maupun sebagai teoritis.

## **BAB II**

# **PERANCANGAN SISTEM DAN TEORI PENUNJANG**



## BAB II

### PERANCANGAN SISTEM DAN TEORI PENUNJANG

Dalam bab ini akan dijelaskan mengenai perancangan sistem yang terdiri dari perancangan perangkat keras dan perangkat lunak beserta teori-teori penunjang yang berhubungan.

#### 2.1 Perancangan Perangkat Lunak

Pada tahap perancangan sistem perangkat lunak ini telah dilakukan beberapa percobaan untuk menentukan metode mana yang lebih baik dan lebih sesuai untuk diaplikasikan pada sistem ini. Pada sistem ini terdapat 2 bagian program yaitu, program pada *microcontroller*, dan program pada PC.

##### 2.1.1 Program pada Mikrokontroler

Program pada *microcontroller* ini didesain dengan tujuan untuk menerima perintah dari PC dan menjalankan perintah-perintah tersebut melalui port serial yang dimiliki. Program ini dibuat menggunakan Code Vision AVR v2.

CodeVisionAVR merupakan software C- cross compiler, dimana program dapat ditulis menggunakan bahasa-C. Software ini mendukung sistem download secara ISP (In-System Programming), yaitu dapat menuliskan program secara langsung pada chip mikro yang dipakai.

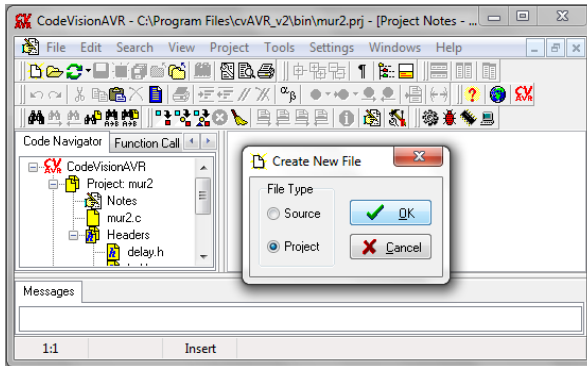
Pembuatan project baru pada CodeVisionAVR dapat dilakukan dengan :

- Install dan jalankan Software CodeVisionAVR  
Klik ganda pada icon >>



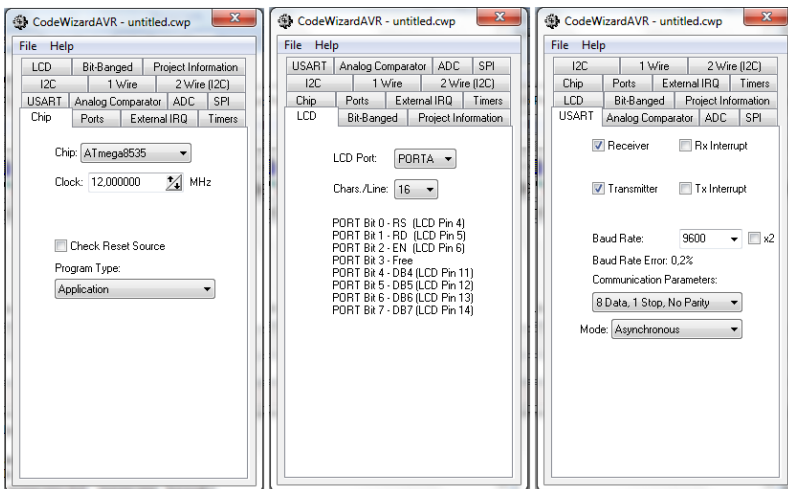
**Gambar 2.1** Icon CVAVR

- Selanjutnya akan tampil jendela awal dari code vision AVR
- Klik menu File >> **New**
- Pilih File Type Project dan **OK**
- Kemudian muncul kotak dialog apakah akan menggunakan Code Wizard AVR untuk mempermudah merancang kerangka program. Pilih **YES**



**Gambar 2.2** Tampilan Awal CodeVisionAVR

- Atur properti yang ada pada jendela wizard tersebut sesuai dengan chip mikro yang digunakan, misal  
 Pada tab 'Chip' pilih Atmega8535 dengan clock 12MHz  
 Pada tab 'LCD' pilih PORT A  
 Pada tab 'USART' tandai Receiver dan Transmitter



**Gambar 2.3** Konfigurasi chip pada CodeWizardAVR

- Kemudian pilih menu **File >> Generate, Save and Exit** untuk menyimpan project dan memulai membuat program.

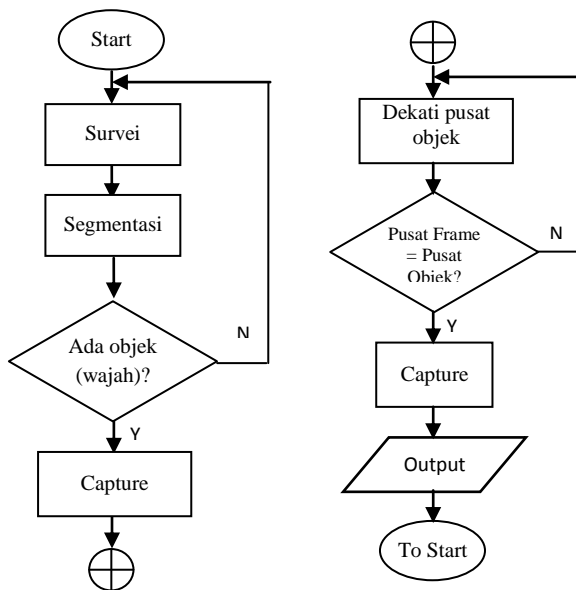


### 2.1.2 Program pada PC (Komputer)

Sedangkan Program untuk PC, didisain agar PC dapat mengambil gambar dari kamera dan mengolah gambar tersebut untuk diambil informasi tentang objek yang dimaksud. Setelah mendapatkan informasi, program akan mengambil keputusan dan memberikan perintah kepada kontroler untuk melakukan langkah berikutnya.

Program pada PC ini merupakan aplikasi konsol deteksi wajah yang dibuat dengan software MS Visual Studio ditambah library OpenCV dan menggunakan metode Viola-Jones.

Flowchart dan penjelasan programnya adalah sebagai berikut :



**Gambar 2.4** Flowchart program

Keterangan :

- Start, robot mulai beraksi.
- Survei, robot memeriksa lingkungan sekitar dengan mengarahkan kameranya.
- *Object segmentation*, robot memeriksa setiap frame image yang masuk untuk diperiksa dan dicari objek yang dimaksud.

- Robot memeriksa apakah pada frame tersebut ada objek wajah. Bila tidak, program kembali ke state survei. Dan bila ada, program dilanjutkan ke step berikutnya.
- Kemudian, program akan mencari titik pusat dari objek yang ditemukan dengan beberapa perhitungan.
- Robot akan mengarahkan bidikan ke titik pusat dari objek.
- Program memeriksa apakah titik pusat dari objek sudah sama dengan titik pusat dari frame kamera.
- Bila belum, robot akan terus memperbaiki bidikan. Sedangkan bila sudah, robot akan melanjutkan ke langkah berikutnya.
- Karena objek telah terbidik, robot akan meng-capture objek dan disimpan.
- Program kembali ke start.

### 2.1.2.1 Open CV Library

**Open CV** adalah singkatan dari Open Computer Vision, yaitu library-library opensource yang di khususkan untuk melakukan image processing. Tujuannya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. Library ini dibuat untuk bahasa C/C++ sebagai optimasi realtime aplikasi, mempunyai API (Application Programming Interface) untuk High level maupun low level, terdapat fungsi-fungsi yang siap pakai untuk loading, saving, akuisisi gambar dan video.

Pada library OpenCV ini mempunyai feature sebagai berikut :

- Manipulasi data gambar (alokasi memori, melepaskan memori, kopi gambar, setting serta konversi gambar)
- Image/Video I/O (Bisa menggunakan camera yang sudah didukung oleh library ini)
- manipulasi matrix dan vektor serta terdapat juga routines linear algebra (products, solvers, eigenvalues, SVD)
- Image processing dasar (filtering, edge detection, pendeteksian tepi, sampling dan interpolasi, konversi warna, operasi morfologi, histograms, image pyramids)
- Analisis struktural
- kalibrasi kamera
- Pendeteksian gerak
- pengenalan objek
- Basic GUI (Display gambar/video, mouse/keyboard kontrol, scrollbar)
- Image Labelling (line, conic, polygon, text drawing)

OpenCV Library dapat diunduh secara gratis di situs resmi <http://sourceforge.net/projects/opencvlibrary>. Berikut ini adalah cara menambahkan librari OpenCV pada lingkungan MS Visual Studio.

- Install Open CV\_v1



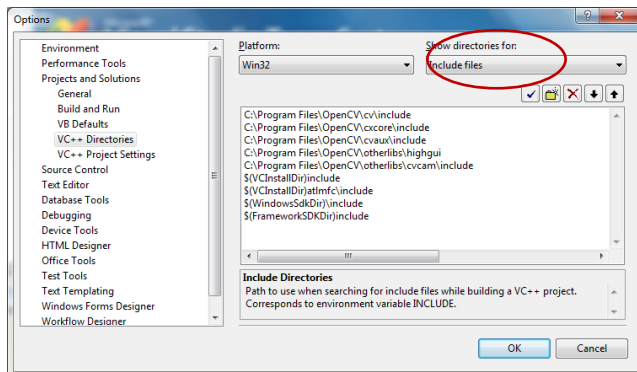
**Gambar 2.5** Icon OpenCV installer

- Pilih *Next* sampai proses instalasi selesai
- Jalankan Aplikasi MS Visual Studio



**Gambar 2.6** Icon Aplikasi MS Visual Studio

- Klik menu *Tools >> Options >> Project and Solutions >> VC++ Directories*
- Pada kolom *Show directories for : Include files*, tambahkan
  - o *C:\Program Files\OpenCV\cv\include\*
  - o *C:\Program Files\OpenCV\cxcore\include\*
  - o *C:\Program Files\OpenCV\cv\include\*
  - o *C:\Program Files\OpenCV\otherlibs\highgui\*
  - o *C:\Program Files\OpenCV\otherlibs\cvcam\include\*



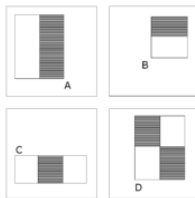
**Gambar 2.7** Konfigurasi OpenCV pada MS Visual Studio

- Dan pada kolom *Show directories for : Library Files*, tambahkan
  - o *C:\Program Files\OpenCV\lib\*
- Ok

### 2.1.2.2 Metode Viola-Jones

Proses deteksi adanya citra wajah dalam sebuah gambar pada OpenCV, menggunakan sebuah metoda yang dipublikasikan oleh Paul Viola dan Michael Jones tahun 2001. Umumnya disebut metoda Viola-Jones. Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat konsep utama :

- Fitur segi empat sederhana yang disebut fitur Haar.
- Integral image untuk pendeteksian fitur secara cepat.
- Metoda machine learning AdaBoost.
- Pengklasifikasi bertingkat (Cascade classifier) untuk menghubungkan banyak fitur secara efisien.



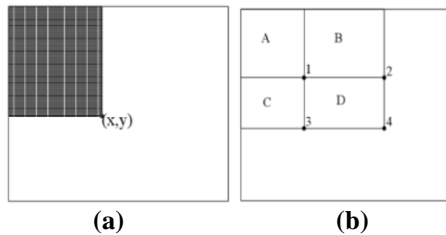
**Gambar 2.8** Contoh fitur pada yang digunakan Viola-Jones

Fitur yang digunakan oleh Viola dan Jones didasarkan pada Wavelet Haar. Wavelet Haar adalah gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah ). Untuk dua dimensi, satu terang dan satu gelap. Selanjutnya kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik disebut **fitur Haar**, atau **fitur Haarlike**, seperti pada gambar 1 di atas menunjukkan fitur yang digunakan dalam pendeteksian citra wajah oleh viola dan jones.

Adanya fitur Haar ditentukan dengan cara mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang. Jika nilai perbedaannya itu diatas nilai ambang atau threshold, maka dapat dikatakan bahwa fitur tersebut ada.

Untuk menentukan ada atau tidaknya dari ratusan fitur Haar pada sebuah gambar dan pada skala yang berbeda secara efisien, Viola dan Jones menggunakan satu teknik yang disebut **Integral Image**. Pada umumnya, pengintegrasian tersebut berarti menambahkan unit-unit kecil secara bersamaan. Dalam hal ini unit-unit kecil tersebut adalah nilai-nilai piksel. Nilai integral untuk masing-masing piksel adalah jumlah dari semua piksel-piksel dari atas sampai bawah. Dimulai dari kiri atas

sampai kanan bawah, keseluruhan gambar itu dapat dijumlahkan dengan beberapa operasi bilangan bulat per piksel.



**Gambar 2.9** Integral image

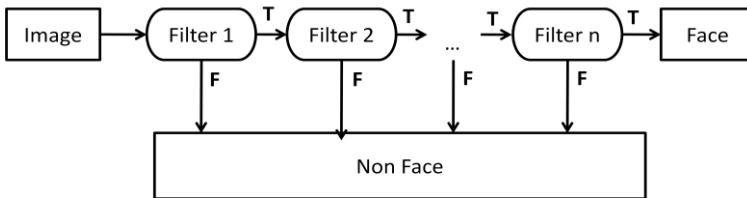
Seperti yang ditunjukkan oleh gambar 2.9(a) di atas setelah pengintegrasian, nilai pada lokasi piksel (x,y) berisi jumlah dari semua piksel di dalam daerah segiempat dari kiri atas sampai pada lokasi (x,y) atau daerah yang diarsir. Untuk menentukan nilai rata-rata piksel pada area segiempat (daerah yang diarsir) ini dapat dilakukan hanya dengan membagi nilai pada (x,y) oleh area segiempat.

Untuk mengetahui nilai piksel untuk beberapa segiempat yang lain misal, seperti segiempat D pada gambar 2.9(b) di atas dapat dilakukan dengan cara menggabungkan jumlah piksel pada area segiempat  $A+B+C+D$ , dikurangi jumlah dalam segiempat  $A+B$  dan  $A+C$ , ditambah jumlah piksel di dalam A. Dengan,  $A+B+C+D$  adalah nilai dari integral image pada lokasi 4,  $A+B$  adalah nilai pada lokasi 2,  $A+C$  adalah nilai pada lokasi 3, dan A pada lokasi 1. Sehingga hasil dari D dapat dikomputasikan  $D = (A+B+C+D) - (A+B) - (A+C) + A$ .

Untuk memilih fitur Haar yang spesifik yang akan digunakan dan untuk mengatur nilai ambangnya (threshold), Viola dan Jones menggunakan sebuah metode machine learning yang disebut AdaBoost. AdaBoost menggabungkan banyak classifier lemah untuk membuat sebuah classifier kuat. Lemah disini berarti urutan filter pada classifier hanya mendapatkan jawaban benar lebih sedikit. Jika keseluruhan classifier lemah digabungkan maka akan menjadi classifier yang lebih kuat. AdaBoost memilih sejumlah classifier lemah untuk disatukan dan menambahkan bobot pada setiap classifier, sehingga akan menjadi classifier yang kuat.

Viola Jones menggabungkan beberapa AdaBoost classifier sebagai rangkaian filter yang cukup efisien untuk menggolongkan

daerah image. Masing-masing filter adalah satu AdaBoost classifier terpisah yang terdiri classifier lemah atau satu filter Haar.



**Gambar 2.10** Cascade classifier

Threshold yang dapat diterima untuk masing-masing level filter di set rendah. Selama proses pemfilteran, bila ada salah satu filter gagal untuk melewati sebuah daerah gambar, maka daerah itu langsung digolongkan sebagai bukan wajah. Namun ketika filter melewati sebuah daerah gambar dan sampai melewati semua proses filter yang ada dalam rangkaian filter, maka daerah gambar tersebut digolongkan sebagai wajah. Viola dan Jones memberi sebutan cascade. Urutan filter pada cascade ditentukan oleh bobot yang diberikan AdaBoost. Filter dengan bobot paling besar diletakkan pada proses pertama kali, bertujuan untuk menghapus daerah gambar bukan wajah secepat mungkin.

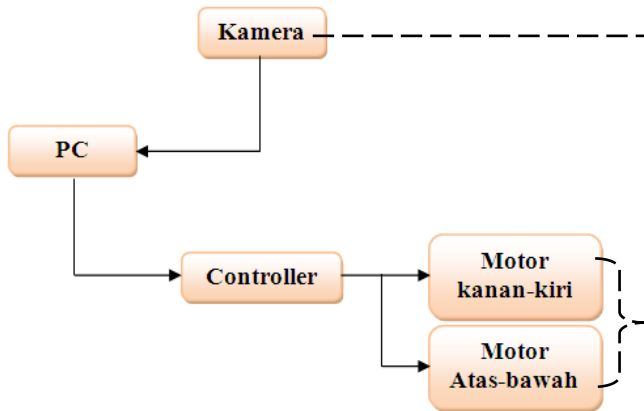
## 2.2 Perancangan Perangkat Keras

Dalam tahap perancangan perangkat keras ini, dititik-beratkan pada pembuatan desain mekanik robot, pembuatan rangkaian minimum sistem dan penempatan peralatan penyusunnya.

Desainnya ini diilustrasikan seperti robot yang memiliki kepala dengan satu mata. Robot ini harus bisa mengarahkan kepala dalam hal ini adalah kamera, ke arah kanan-kiri dan ke arah atas-bawah. Robot ini tidak dapat berpindah tempat, namun sebisa mungkin didesain untuk bisa ditempatkan dimana saja.

Selain Desain, harus dipertimbangkan juga bahan utama penyusun robot. Bahan yang dipakai untuk desain tersebut haruslah bahan yang mudah dibentuk, ringan, tetapi kuat, disini kami menggunakan bahan kaca acrylic transparan yang memiliki ketebalan 3mm.

Berikut ini adalah diagram penyusun sistem EyeBot :



**Gambar 2.11** Diagram penyusun EyeBot

Gambar 2.11 diatas menunjukkan perlengkapan-perengkapan penyusun yang dimiliki oleh sistem untuk dapat menjalankan tugasnya dengan baik. Dapat kita lihat, pada sistem terdapat sebuah kamera. Kamera pada sistem ini berfungsi sebagai mata. Tugas mata adalah untuk melihat daerah sekeliling dan melaporkannya ke PC.

Selain kamera, pada sistem juga terdapat PC. Tugas dari PC adalah sebagai pengambil keputusan. Dari laporan yang diberikan kamera, PC akan mempertimbangkan langkah berikutnya yang akan diambil. Keputusan sepenuhnya berada di tangan PC yang juga dapat disebut sebagai "otak".

Untuk langkah berikutnya, PC tidak bekerja sendiri. PC mengendalikan peralatan lain melalui rangkaian controller. PC hanya perlu memberikan perintah melalui kabel serial. Tugas rangkain controller pada sistem adalah sebagai pelaksana perintah yang diberikan oleh PC. Untuk kemudian, rangkain controller dapat mengendalikan peralatan lain yaitu motor

Fungsi motor pada sistem ini adalah sebagai penggerak kamera, sehingga sistem dapat terus memantau sebuah target.

### 2.2.1 Kamera

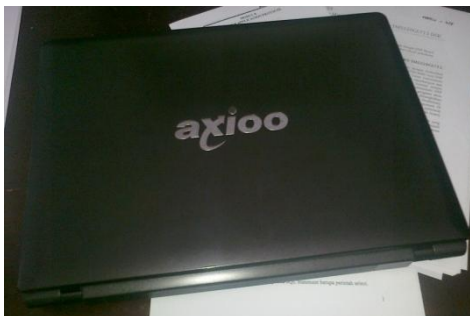


**Gambar 2.12** Webcam

Pada sistem ini, kamera yang digunakan adalah kamera web, atau yang sering disebut webcam. Untuk pemilihan kamera, sebaiknya kita memilih webcam yang memiliki kecepatan serta resolusi yang bagus dan juga harus memiliki kemampuan yang baik dalam adaptasi terhadap cahaya di sekelilingnya.

### 2.2.2 Personal Computer (PC)

PC atau Personal Computer adalah seperangkat unit pemroses data yang terdiri dari monitor, keyboard, motherboard, processor, dan lain-lain. Fungsi PC pada sistem ini adalah sebagai "otak" yang harus mengambil keputusan untuk langkah berikut yang harus diambil. PC ini menerima informasi dari webcam yang berbentuk image yang merepresentasikan situasi di sekeliling sistem. Data ini dikirim oleh webcam melalui jalur USB yang dimiliki oleh PC.



**Gambar 2.13** Laptop sebagai pengganti PC

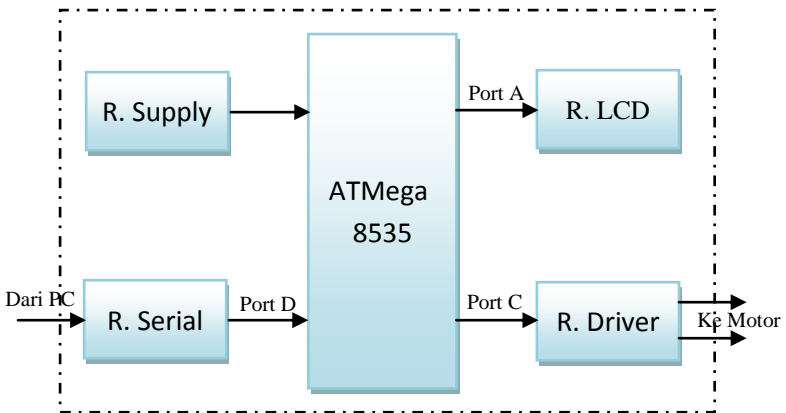


Selain itu, PC juga harus mengirimkan perintah ke rangkaian kontroler. PC tidak dapat melakukan pengendalian terhadap perangkat luar sendiri. Tetapi, semua keputusan yang diambil itu berada ditangan PC, seperti kemana kamera harus bergerak, kapan harus meng-*capture*, dan lain sebagainya.

### 2.2.3 Minimum System

Minimum system adalah sebuah rangkaian yang terdiri dari microcontroller dan beberapa sub rangkaian lain yang terhubung pada port yang terdapat pada microcontroller.

Minimum sistem ini bertugas menerima perintah dari Personal Computer ( PC ) dan melaksanakannya. MinSys ini menerima perintah melalui komunikasi serial yang dilewatkan rangkaian serial.



**Gambar 2.14** Diagram Minimum System ATmega 8535

Sedangkan untuk melaksanakan perintah, MinSys ini menggunakan rangkaian driver motor untuk menggerakkan 2 buah motor. Motor-motor yang dimaksud adalah motor penggerak kamera nantinya, dan untuk melaksanakannya, MinSys akan mengatur nilai pada port yang terkait.

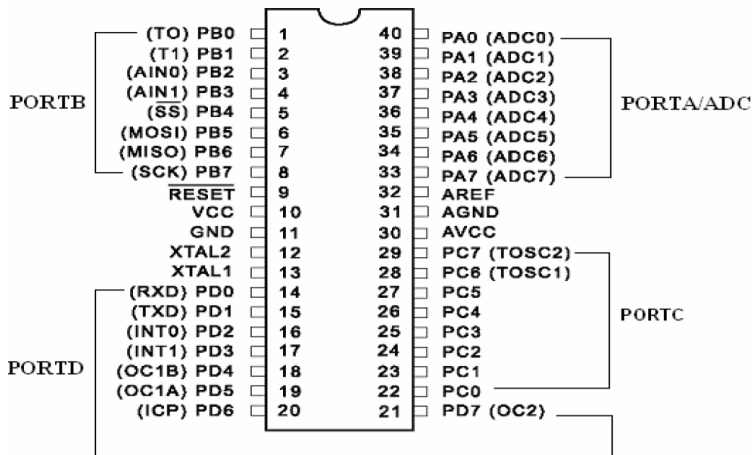
#### 2.2.3.1 Kontroler ATmega 8535

Kontroler Atmega 8535 merupakan mikrokontroler 8-bit AVR dengan Kapasitas memory maksimum sebesar 16 Kbytes yang tersimpan didalam *System Programmable Flash*-nya. Atmega 8535 merupakan chip IC produksi ATMEL yang termasuk golongan *single*

*chip microcontroller*, dimana semua rangkaian termasuk memori dan I/O tergabung dalam satu pak IC. Dalam pemrogramannya kontroller ini dapat dijalankan menggunakan 2 bahasa yaitu bahasa Assembly atau bahasa C. Sehingga memungkinkan pengguna dapat mengoptimalkan kinerja sistem yang dibuat secara fleksibel.

### 2.2.3.1.1 Konfigurasi Pin ATMega 8535

IC Atmega 8535 ada 2 jenis yaitu jenis PDIP (berbentuk balok) dan jenis TQFP/MLF (berbentuk kotak) yang pada dasarnya memiliki fasilitas yang sama, hanya saja memiliki bentuk yang berbeda sehingga letak kaki-kaki IC berbeda mengikuti bentuknya.



**Gambar 2.15** Konfigurasi Pin Out ATMega 8535

Dari gambar diatas dapat dilihat ada 40 kaki atau pin IC yang memiliki fungsi sebagai berikut:

1. **V<sub>cc</sub>**  
Sebagai sumber tegangan digital.
2. **GND**  
Sebagai Ground.
3. **Port A (PA7 ... PA0)**  
Port A berfungsi sebagai input analog untuk A/D konverter. Selain itu sebagai I/O port 8 bit *bi-directional* jika A/D Konv. tidak digunakan.

**4. Port B (PB7 ... PB0)**

Port B berfungsi sebagai I/O port 8 bit *bi-directional*.

**5. Port C (PC7 ... PC0)**

Port C berfungsi sebagai I/O port 8 bit *bi-directional*. Jika *JTAG interface* diaktifkan, maka *pull-up* resistor pada pin PC5, PC3 dan PC2 akan ikut aktif meskipun reset sedang berlangsung.

**6. Port D (PD7 ... PD0)**

Port D berfungsi sebagai I/O port 8 bit *bi-directional*.

**7. RESET**

Adalah masukan reset (aktif low). Pulsa transisi dari tinggi ke rendah akan me-reset.

**8. XTAL1**

Sebagai input untuk *inverting oscillator* dan input untuk *internal clock operating circuit*.

**9. XTAL2**

Adalah Output dari *Inverting oscillator amplifier*.

**10.  $AV_{CC}$** 

$AV_{CC}$  adalah sumber tegangan pin untuk port A dan A/D konverter

**11.  $A_{REF}$** 

$A_{REF}$  adalah pin referensi analog untuk A/D konverter.

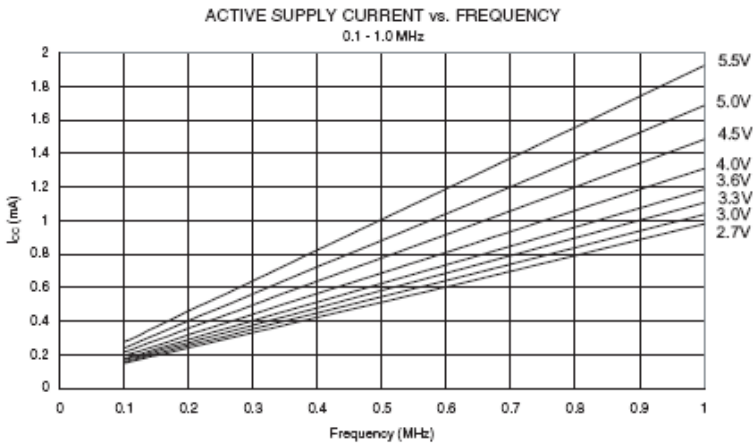
**2.2.3.1.2 Karakteristik ATmega 8535**

Untuk menjaga agar sistem dapat memiliki kinerja yang lebih lama, maka perlu diperhatikan beberapa factor, diantaranya karakteristik dari Atmega 8535. Berikut ini beberapa karakteristik Atmega 8535 yang telah ditabelkan.

**Tabel 2.1** Karakteristik Elektrik ATmega 32

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on any Pin except RESET	
With respect to ground.....	-0.5V to Vcc +0.5V
Voltage on RESET with respect to Ground.....	-0.5V to +13V
Maximum Operating Voltage.....	6V
DC Current per I/O Pin.....	40mA
DC Current Vcc and GND Pins.....	200mA PDIP and 400mA TQFP/MLF

Adapun karakteristik antara arus  $I_{cc}$  dengan frekuensi yang digrafikkan sebagai berikut:



**Gambar 2.16** Grafik respon antara  $I_{cc}$  dengan  $f$

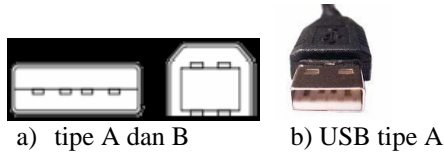
### 2.2.3.2 Metode Komunikasi

Beragamnya metode komunikasi antarmuka memungkinkan untuk menggunakan metode komunikasi yang tepat dan sesuai dengan sistem yang akan dibangun. Salah satu metode komunikasi yang digunakan dan yang akan dijelaskan berikut ini adalah USB 2.0 dan RS232.

#### 2.2.3.2.1 Komunikasi Antara Kamera Dengan PC

Metode komunikasi yang akan diterapkan pada antarmuka PC-Kamera adalah USB 2.0 sebagai standar bus berseri untuk perangkat penghubung. Dalam perkembangan teknologi piranti komunikasi, USB adalah suatu teknologi komunikasi data serial yang bersifat multi guna dengan topologi BUS. USB 1.0 memiliki kecepatan sekitar 12 Mbps, sedangkan USB 2.0 memiliki kecepatan sekitar 480 Mbps.

Sistem USB mempunyai desain yang asimetris, yang terdiri dari pengontrol host dan beberapa peralatan terhubung yang berbentuk pohon dengan menggunakan peralatan hub yang khusus.



**Gambar 2.17** Jenis konektor USB

Konektor USB memiliki 4 lempeng tembaga atau emas yang masing-masing lempeng/kaki memiliki karakteristik berbeda-beda, yaitu,

1. Tegangan Bus yang memiliki range antara 4.75V-5.25V
2. Transfer data negatif (D -)
3. Transfer data positif (D +)
4. Lempeng Pentanahan (GND)



**Gambar 2.18** Penetapan Kaki Konektor

#### 2.2.3.2.2 Komunkasi Antara PC Dengan Kontroler

Pada prinsipnya, komunikasi serial ialah komunikasi dimana pengiriman data dilakukan per bit, sehingga lebih lambat dibandingkan komunikasi parallel seperti pada port printer yang mampu mengirim 8 bit sekaligus dalam sekali detak. Beberapa contoh komunikasi serial ialah mouse, scanner dan system akuisisi data yang terhubung ke port COM1/COM2.

Peralatan pada komunikasi serial port dibagi menjadi 2 (dua) kelompok yaitu *Data Communication Equipment (DCE)* dan *Data Terminal Equipment (DTE)*. Contoh dari DCE ialah modem, plotter, scanner dan lain lain sedangkan contoh dari DTE ialah terminal di komputer. Spesifikasi elektronik dari serial port merujuk pada *Electronic Industry Association (EIA)*:

1. “Space” (logika 0) ialah tegangan antara + 3 hingga +25 V.
2. “Mark” (logika 1) ialah tegangan antara –3 hingga –25 V.
3. Daerah antara + 3V hingga –3V tidak didefinisikan /tidak terpakai
4. Tegangan open circuit tidak boleh melebihi 25 V.
5. Arus hubungan singkat tidak boleh melebihi 500mA.

Komunikasi serial membutuhkan port sebagai saluran data. Berikut tampilan port serial DB9 yang umum digunakan sebagai port serial



**Gambar 2.19** Port DB9 jantan

Konektor port serial terdiri dari 2 jenis, yaitu konektor 25 pin (DB25) dan 9 pin (DB9) yang berpasangan (jantan dan betina). Bentuk dari konektor DB-25 sama persis dengan port paralel. Umumnya COM1 berada dialamat 3F8H, sedangkan COM2 dialamat 2F8H.

**Tabel 2.2** Jenis Sinyal RS232

Nama Sinyal	Arah Sinyal	Nomor Kaki Konektor	
		DB9	DB25
Signal Common	-	5	7
Transmitted Data (TD)	Ke DCE	3	2
Received Data (RD)	Dari DCE	2	3
Request to Send (RTS)	Ke DCE	7	4
Clear to Send (CTS)	Dari DCE	8	5
DCE Ready (DSR)	Dari DCE	6	6
DTE Ready (DTR)	Ke DCE	4	20
Ring Indicator (RI)	Dari DCE	9	22
Data Carrier Detect (DCD)	Dari DCE	1	8

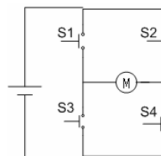
**2.2.4 Motor DC**

Motor DC merupakan sebuah elektrik motor yang menggunakan tegangan DC yang mengkonversikan besaran listrik menjadi besaran mekanik. Motor secara umum terbagi atas dua macam yaitu motor arus bolak balik (*electromagnetic alternating current motor*) yang biasa disebut dengan motor AC dan motor arus searah (*electromugnetic direct current motor*) yang biasa disebut dengan motor DC. Motor DC yang pada umumnya digunakan pada pekerjaan yang kecil dan lebih cocok untuk digunakan pada aplikasi - aplikasi elektronika misalnya robot mobil. Motor DC ini mempunyai dua terminal elektrik.

Dengan memberikan beda tegangan pada kedua terminal tersebut maka motor akan dapat berputar pada satu arah dan apabila polaritas dari tegangan tersebut dibalik, maka arah putaran motor akan terbalik pula. Hal ini berlaku pada motor DC dan tidak berlaku pada motor AC. Polaritas dari tegangan yang diberikan pada dua terminal menentukan arah putaran motor sedangkan beda tegangan yang diberikan menentukan kecepatan motor tersebut.

Pada interfacing motor DC memerlukan *supply* arus yang cukup besar, untuk itu diperlukan suatu rangkaian *interfacing* antara mikrokontroller port dengan motor untuk mendapat *supply* arus yang cukup. *Interfacing* ini dapat diimplementasikan dengan berbagai macam komponen, antara lain : *relay*, *bipolar transistor*, *power mosfet* dan motor driver *integrated circuit*.

Berbagai macam komponen tersebut adalah rangkaian yang menjadi dasar membentuk *motor driver*. Rangkaian ini yang disebut dengan *H-Bridge* yang terdiri dari empat *switch* yang terhubung dengan topology H seperti pada gambar 2.20 berikut ini :



**Gambar 2.20** Topologi H-Bridge

Dari gambar 2.20, jika S 1 dan S4 ditutup sementara S2 dan S3 terbuka, maka arus yang mengalir pada motor dari kiri ke kanan, dan jika S2 dan S3 ditutup sementara S1 dan S4 terbuka, maka arus mengalir terbalik dari yang semula yaitu dari kanan ke kiri. Sehingga putaran motor pun dapat berubah arah sesuai dengan arus yang melewatinya. Jika semua teminal dibiarkan mengambang maka poros motor akan bebas dan bila semua terminal terhubung maka motor akan berhenti atau tertahan.

Motor yang dipilih adalah motor yang memiliki *gearbox*, sehingga mampu menggerakkan benda-benda yang cukup berat.



**Gambar 2.21** Motor DC dan Gearbox

\*\*\*halaman ini sengaja dikosongkan\*\*\*



## **BAB III**

# **PEMBUATAN SISTEM DAN PENGUJIAN DAN ANALISA**



## BAB III

### PEMBUATAN SISTEM PENGUJIAN DAN ANALISA

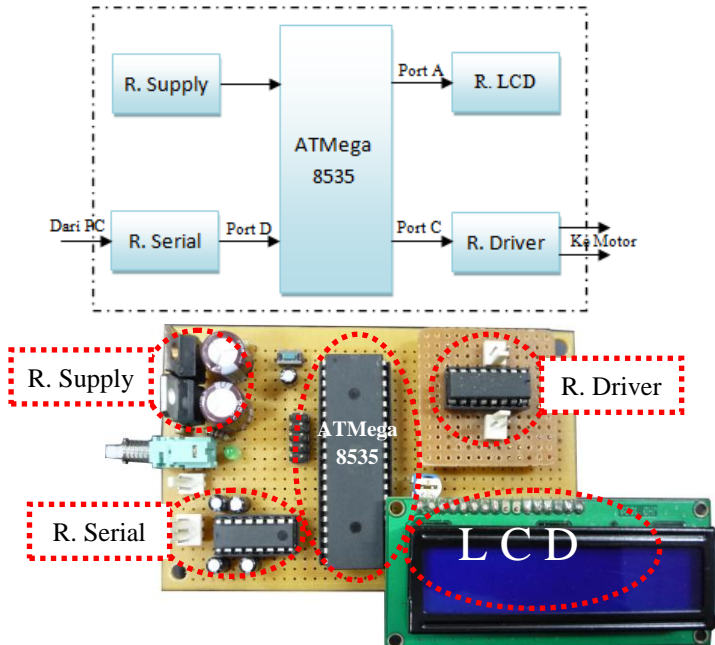
Pada bab ini akan dijelaskan langkah-langkah pembuatan sistem beserta pembahasannya. Pembuatan sistem ini meliputi pembuatan hardware dan software, dilanjutkan dengan pengujiannya.

#### 3.1 Pembuatan Sistem

##### 3.1.1 Pembuatan Perangkat Keras

Berikut ini akan dijelaskan secara lebih detail pembuatan perangkat keras yang meliputi *minimum system* dan konstruksi robot.

##### 3.1.1.1 *Minimum System*



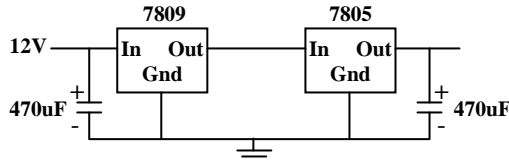
**Gambar 3.1** Diagram dan Rangkaian *Minimum System*

Pada gambar x diatas, ditunjukkan kesesuaian antara rancangan *minimum system* yang berbentuk diagram dengan rangkaian hasil realisasi dari rancangan tersebut.

Pada *minimum system*, terdapat beberapa bagian dengan tugas-tugas tersendiri. Bagian-bagian yang dimaksud diantaranya :

#### A. Rangkaian supply

Rangkaian supply berfungsi untuk menurunkan tegangan dari tegangan baterai yaitu 12V menjadi tegangan logic yaitu 5V.

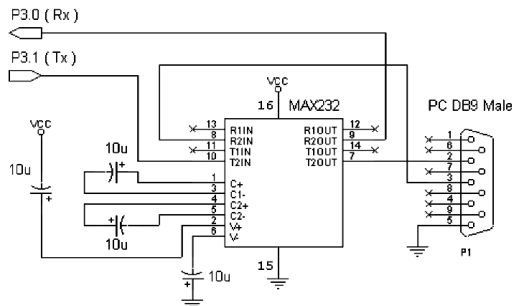


**Gambar 3.2** Schematic rangkaian supply.

Rangkaian ini terdiri dari 2 buah IC regulator dengan tipe 7809 dan 7805. Penggunaan dua buah IC disini dengan tujuan, agar ic 7805 tidak bekerja terlalu berat sehingga menghasilkan panas yang berlebihan.

#### B. Rangkaian serial

Rangkaian ini berfungsi untuk membantu Microcontroller untuk berkomunikasi serial dengan peralatan yang terhubung dengan *minimum system*.

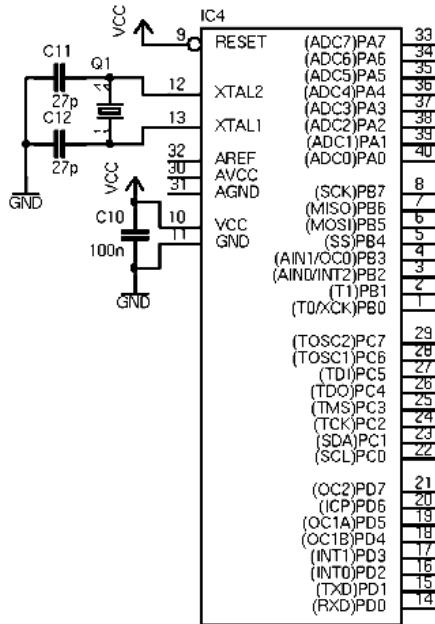


**Gambar 3.3** Schematic rangkaian serial.

Komponen inti penyusun rangkaian ini adalah sebuah IC MAX232. IC ini dirangkai dengan beberapa resistor sebagai pengaman. Jalur komunikasi ini memiliki 3 kabel, yaitu Tx, Rx dan Gnd. Rangkaian ini terhubung dengan *microcontroller* pada port yang memiliki fungsi sebagai Tx, Rx, dan pada port Ground dari *microcontroller*.

### C. Kontroler ATmega 8535

Kontroler ATmega 8535 disini bertugas sebagai koordinator dari rangkaian yang terpasang pada port-portnya. uC ini memiliki wewenang untuk mengendalikan motor dan menerima perintah dari PC melalui jalur data serial.

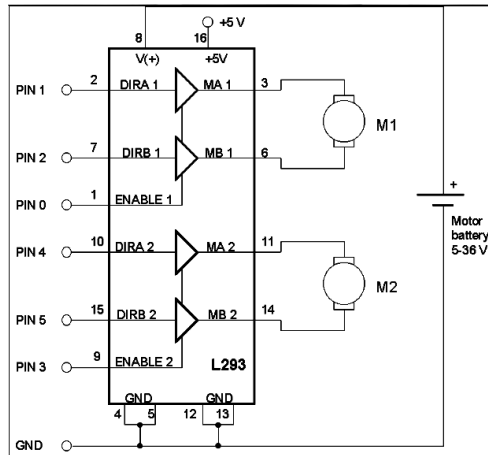


**Gambar 3.4** Schematic rangkaian ATmega 8535

Untuk dapat bekerja, uC ini membutuhkan sebuah rangkaian clock yang terdiri dari sebuah crystal dan dua buah kapasitor, serta sebuah rangkaian supply. Rangkaian clock disini berperan sebagai jantung yang memproduksi denyut atau clock.

### D. Driver motor

Driver motor ini berfungsi untuk mengendalikan dua buah motor. Dengan driver ini motor dapat digerakkan ke dua arah dengan cara mengatur nilai dari kaki-kaki dari driver.

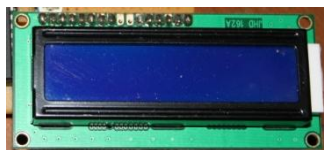


**Gambar 3.5** *Schematic driver motor*

Rangkaian ini terdiri dari sebuah IC driver tipe L293D. IC ini memang didedikasikan sebagai IC driver motor. Untuk dapat bekerja, IC ini cukup diberikan tegangan. IC ini membutuhkan 2 jenis tegangan. Tegangan untuk motor dan tegangan logic. Tegangan logic cukup diberi 5V. Sedangkan untuk tegangan motor dapat diberi tegangan bervariasi dari 9V – 24V.

## E. Modul LCD

Untuk mempermudah mengetahui arah gerak motor serta status-status dari robot, disediakan sebuah LCD. LCD yang digunakan ini adalah sebuah modul LCD. Dengan adanya modul ini akan mempermudah untuk menampilkan tulisan-tulisan.

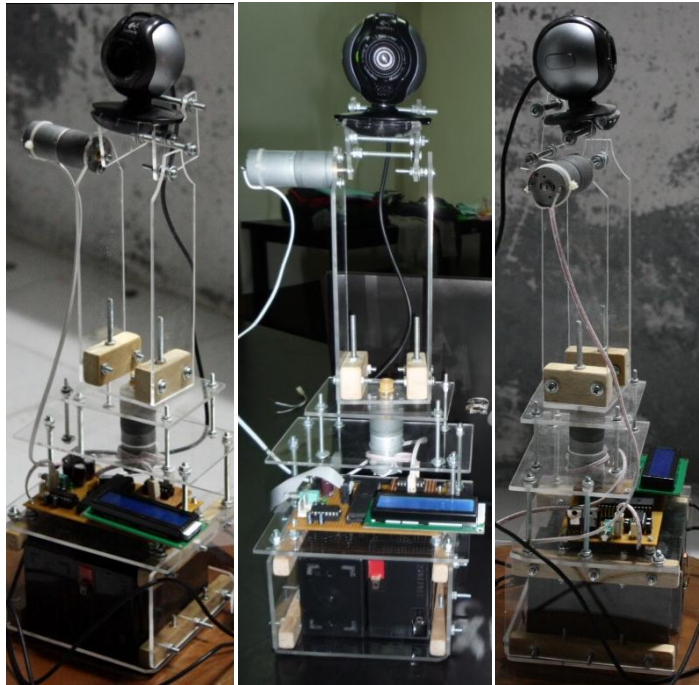


**Gambar 3.6** Modul LCD 16x2

Cara merangkai modul LCD ini ke ATmega 8535 adalah dengan menghubungkan fungsi kaki yang bersesuaian antara modul LCD dengan kaki-kaki dari ATmega 8535.

### 3.1.1.2 Konstruksi Sistem

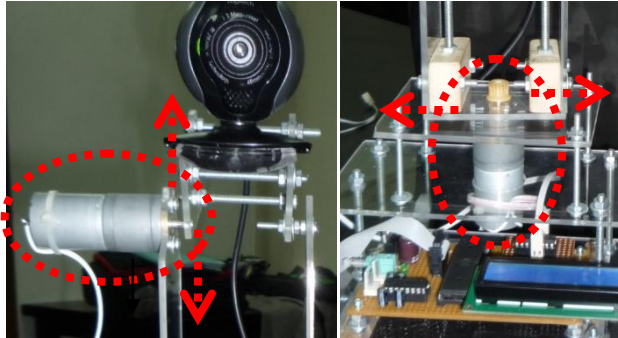
Pada bagian konstruksi robot ini, kita akan membahas lebih dalam ke bagian mekanik dari gunnerbot dan cara membangunnya. Pada dasarnya, robot ini dibangun dari sebuah *acrylic* setebal 3 mm yang dipotong-potong sehingga membentuk badan robot seperti yang berikut ini:



**Gambar 3.7** Konstruksi Robot

Dengan konstruksi robot seperti diatas, memungkinkan robot dapat terus memantau target. Hal itu dikarenakan robot dapat memutar kamera secara *horizontal* dan *vertical*.

Seperti yang pernah kita singgung sebelumnya, mekanik robot ini digerakkan oleh dua buah motor DC ber-*gearbox*. Berikut adalah tempat pemasangan dari kedua motor tersebut :



**Gambar 3.8** Tempat pemasangan 2 buah motor

Pada gambar diatas, nampak pada motor sebelah kiri terpasang secara miring. Dengan cara pemasangan seperti ini, memungkinkan motor untuk menggerakkan kamera ke arah *tilt up-down* (atas-bawah).

Berbeda dengan motor yang terlihat disebelah kanan gambar. Motor itu terpasang pada posisi berdiri sehingga dapat menggerakkan kamera dan senjata ke arah *pan* kanan-kiri. Dengan kombinasi gerakan dari kedua buah motor tersebut, robot dapat melakukan manuver-manuver untuk membidik target.

### 3.1.2 Pembuatan Software

Selain pembuatan perangkat keras, untuk membangun sistem ini kita juga perlu program yang menjalankannya. Pembuatan perangkat lunak atau program ini dibagi menjadi dua bagian. Bagian pertama adalah kode program yang berjalan di komputer. Sedangkan bagian kedua adalah program yang akan di-*download* ke *microcontroller* yang bertugas sebagai pelaksana perintah dari komputer.

Program yang berjalan di komputer ini meliputi pengaktifan kamera untuk mendapatkan gambar, perhitungan posisi, dan pengiriman data serial. Sedangkan untuk program yang di-*download* ke *microcontroller* meliputi penerimaan perintah dari PC, kemudian melaksanakan perintah tersebut.



### 3.1.2.1 Mengaktifkan kamera

Langkah pertama dari program pada PC adalah mengaktifkan kamera, tujuannya agar PC mendapatkan gambar dari kamera tersebut dan ditampilkan pada layar. Berikut ini potongan programnya :

```
main()
{
    cvCapture* capture = 0;
    IplImage *frame, *frame_copy = 0;
    capture = cvCaptureFromCAM(0);
    frame = cvRetrieveFrame( capture );
    frame_copy = cvCreateImage(cvGetSize(pvideoFrame), 8, 3);
    cvNamedWindow( "E Y E B O T", 1 );
    cvShowImage( "E Y E B O T", img );
}
```

keluaran dari potongan program di atas adalah sebagai berikut :



**Gambar 3.9** Hasil *capture* dari kamera

### 3.1.2.2 Deteksi wajah

Setelah didapat input frame dari kamera, langkah selanjutnya adalah memeriksa setiap gabungan dari pixel untuk mencari objek wajah yang ada. Proses deteksi wajah tersebut menggunakan metode viola-jones, yaitu dengan cara mencocokkan setiap fitur pada frame dengan contoh wajah yang ada pada sebuah file database berekstensi \*.xml. File tersebut berisi ribuan sampel positif dari wajah.

```
const char* cascade_name = "haarcascade_frontalface_default.xml";
cascade = (CvHaarClassifierCascade*)cvLoad( cascade_name, 0, 0, 0 );
//Apakah file .xml ada?? dicek dulu reeeekkk!!
if( !cascade )
{
    fprintf( stderr, "ERROR: File *.xml tidak ada...\n" );
    return (1);
}
```

Setelah proses load database, setiap frame yang didapat dari kamera tadi dikirimkan ke fungsi yang bertugas untuk menandai objek wajah yang ada.

```
//Looping untuk setiap wajah yg terdeteksi
for( i = 0; i < (faces->total); i++ )
{
    CvRect* r = (CvRect*)cvGetSeqElem( faces, i );
    pt1[i].x = r->x*scale;
    pt2[i].x = (r->x+r->width)*scale;
    pt1[i].y = r->y*scale;
    pt2[i].y = (r->y+r->height)*scale;
    cvRectangle(img,pt1[i],pt2[i],CV_RGB(255,255,0),2,4,0);
}
```



b. Single Object

b. Multiple Object

**Gambar 3.10** Hasil deteksi wajah

### 3.1.2.3 Perhitungan posisi

Yang dimaksud perhitungan posisi di sini adalah pencarian titik pusat dari objek wajah yang ada, karena batas-batas objek yang dimaksud sudah tersedia, maka pencarian titik pusat cukup dengan perhitungan yang sederhana.

```
//Pusat objek
center[i].x = (pt1[i].x+pt2[i].x)/2;
center[i].y = (pt1[i].y+pt2[i].y)/2;
```

Langkah berikutnya adalah membandingkan dan mencari jarak antara titik pusat dari objek dengan titik pusat dari frame. Hal ini dilakukan dengan tujuan untuk mengetahui kemana kamera harus digerakkan.

Sebagai ilustrasi, dapat kita lihat pada tabel berikut:

**Tabel 3.1** Tabel kebenaran aksi robot

Jarak sumbu X = +	<i>Pan</i> kamera ke kanan
Jarak sumbu X = -	<i>Pan</i> kamera ke kiri
Jarak sumbu Y = +	<i>Tilt</i> kamera ke atas
Jarak sumbu Y = -	<i>Tilt</i> kamera ke bawah

Kemudian untuk kemungkinan-kemungkinan lain adalah kombinasinya. Sedangkan kode untuk melakukan kalkulasi tersebut adalah sebagai berikut.

```
//mencari target terdekat
DistTarget[i] = sqrt(pow((center[i].x - CtrFrm.x),2)
                    + pow((center[i].y - CtrFrm.y),2));
if(DistTarget[i] < closetar)
{
    closetar = DistTarget[i];
    cclose = i;
    CtrTargetClose = center[i];
}

// Hitung jarak target
Jarak.x = abs(center[cclose].x - CtrFrm.x);
Jarak.y = abs(center[cclose].y - CtrFrm.y);
```

Karena nantinya akan terdapat kemungkinan adanya banyak objek wajah yang terdeteksi pada frame, maka program di atas akan menentukan satu dari sekian objek yang ada untuk dijadikan target prioritas, yaitu objek yang memiliki jarak terdekat dari pusat frame kamera. Selanjutnya koordinat pusat dari objek terdekat akan disimpan pada variabel Jarak.x dan Jarak.y

### 3.1.2.4 Transmisi Data Serial

Langkah yang berikutnya diambil setelah melakukan pengolahan terhadap citra yang didapat adalah mengomunikasikan keputusan yang diambil dengan *microcontroller*. Untuk menjalin komunikasi tersebut, dibutuhkan metode-metode atau fungsi-fungsi yang dapat melakukan pengiriman data serial dari program ke piranti

yang dituju. Berikut ini adalah beberapa kode penting yang diperlukan untuk menjalin komunikasi serial.

Kode berikut ini adalah kode untuk mengatur port mana yang akan digunakan dan *baudrate* yang akan dipakai.

```
// Serial Port Setting
#define PORTNUM 1
#define BAUDRATE 9600
// -----
```

Dapat kita lihat, port yang digunakan adalah port COM 1 dan *baudrate* diatur pada kecepatan 9600 Kbps. Untuk pengaturan port COM, kita harus menggunakan port yang tidak terpakai. Sedangkan untuk pengaturan *baudrate*, harus disamakan dengan piranti yang dituju.

Selanjutnya, kita harus menginisialisasi dan membuka port yang kita pilih dengan kode berikut ini.

```
// inisialiasasi Port Serial
if((InitPort(PORTNUM,BAUDRATE,&hCom,&dcB))!=0)
{
    cout<<"\nERROR : Port Serial COM"<<PORTNUM<<" Gagal\n";
    return (1);
}
```

Setelah port berhasil terbuka, akan muncul pesan "port serial siap". Sedangkan jika gagal, akan muncul pesan "Gagal membuka Port Com 1".

Mari kita asumsikan port berhasil dibuka. Selanjutnya, yang perlu kita lakukan adalah menyiapkan data yang akan dikirim melalui jalur serial ini. Pengaturan data ini dilakukan pada variabel `writebuffer[]`.

Karena pengaturan data ini harus sesuai dengan keputusan dari program, maka pengaturan ini kita serahkan kepada program dengan memasukkan ke kondisi sebagai berikut.

```
//=====
if(tilt == 'o' && pan == 'o')
{
    writebuffer[1] = 's'; // terkunci
}
else if(tilt == 'o' && pan == '-')
{
    writebuffer[1] = 'a'; // diam , kiri
}
else if(tilt == 'o' && pan == '+')
{
    writebuffer[1] = 'd'; // diam , kanan
}
else if(tilt == '-' && pan == 'o')
{
    writebuffer[1] = 'x'; // bawah , diam
}
else if(tilt == '-' && pan == '-')
{
    writebuffer[1] = 'z'; // bawah , kiri
}
else if(tilt == '-' && pan == '+')
{
    writebuffer[1] = 'c'; // bawah , kanan
}
else if(tilt == '+' && pan == 'o')
{
    writebuffer[1] = 'w'; // atas , diam
}
else if(tilt == '+' && pan == '-')
{
    writebuffer[1] = 'q'; // atas , kiri
}
else if(tilt == '+' && pan == '+')
{
    writebuffer[1] = 'e'; // atas , kanan
}
}
```

Keluaran dari kode tersebut adalah data yang sesuai dengan keadaan yang aktual dan siap dikirim. Kemudian, proses pengiriman data tersebut dapat dilakukan dengan kode berikut ini.

```
//Begin of serial data
writebuffer[0] = '#';
tempo = writebuffer[1];
temp2 = writebuffer[2];
temp3 = writebuffer[3];

// mengirim data serial
writeData(&hCom,4);
```

Fungsi utama pengiriman data serial dari kode diatas adalah writeData(\$hCom,3). Kode tersebut memiliki arti, menulis data pada port yang terbuka sebanyak 3 karakter. Sedangkan karakter yang

dimaksud adalah karakter yang sudah diputuskan dan disiapkan pada kode sebelumnya.

### 3.1.2.5 Program Controller

Seperti yang disebutkan sebelumnya, selain membutuhkan pemrograman di sisi PC, sistem ini juga memerlukan pemrograman di sisi *controller* (*controller side*). Tugas dari program yang berjalan pada sisi *controller* ini adalah untuk saling berkoordinasi dengan PC. Sedangkan data yang dikomunikasikan tersebut menentukan apa yang harus dilakukan oleh bagian *controller*.

Dalam menjalankan tugasnya, program ini dilengkapi beberapa fungsi yang dapat mendukung fungsional dari bagian *controller*. Berikut ini adalah kode program yang merupakan inisialisasi dari port-port yang dipakai. Selain itu, kode ini juga merupakan inisialisasi dari port yang akan dihubungkan dengan LCD.

```
#include <mega8535.h>

//Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>

//Standard Input/output functions
#include <stdio.h>
#include <delay.h>

//enable motor
#define enA PORTB.0
#define enB PORTB.1

//motor A
#define A1 PORTC.0
#define A2 PORTC.1

//motor B
#define B1 PORTC.6
#define B2 PORTC.7
```

Pada kode diatas dinyatakan bahwa PORTC.0 dan PORTC.1 digunakan untuk motor A, dan seterusnya. Selain itu, dinyatakan pula bahwa PORTA ( 0x1B ) digunakan untuk LCD.

Berikutnya adalah fungsi-fungsi yang digunakan untuk mengatur aktifitas motor. Untuk mempermudah dalam pengembangan, maka untuk pengaturan-pengaturan pada motor sengaja kami jadikan fungsi-fungsi. Yang berikut ini adalah fungsi untuk menjalankan motor :

```

void run(int motor, int direction,
        unsigned char speed){

    if(motor == 1){
        if(direction == 0){
            A1 = 0; A2 = 1;
        }
        else if(direction == 1){
            A1 = 1; A2 = 0;
        }
        lpwm = speed;
    }
    else if(motor == 2){
        if(direction == 0){
            B1 = 0; B2 = 1;
        }
        else if(direction == 1){
            B1 = 1; B2 = 0;
        }
        rpwm = speed;
    }
}

```

Tujuan dari fungsi diatas adalah untuk menentukan motor mana yang akan dikontrol, kemana arah putaran motor, dan berapa kecepatan motor yang diinginkan. Fungsi ini juga melibatkan pengaturan PWM. Pengaturan PWM ini bertujuan untuk mengatur kecepatan motor. PWM ini mengatur lebar pulsa *on* dan lebar pulsa *off*. Dengan pengaturan tersebut, tegangan efektif yang masuk ke motor dapat diubah-ubah sehingga dapat mempengaruhi kecepatan motor.

Selain menjalankan motor, kita tentu juga perlu melakukan pengereman atau menghentikan putaran motor. Berikut ini adalah fungsi untuk menghentikan putaran motor :

```

void stop(int motor){
    if(motor == 1){
        lpwm = 250;
        A1 = 1; A2 = 1;
        delay_ms(250);
        lpwm = 0;
        A1 = 0; A2 = 0;
    }
    if(motor == 2){
        rpwm = 250;
        B1 = 1; B2 = 1;
        delay_ms(250);
        rpwm = 0;
        B1 = 0; B2 = 0;
    }
}

```

Pada fungsi diatas, kita cukup menentukan motor mana yang akan dihentikan. Sebagai contoh, bila kita ingin menghentikan motor 1, yang perlu kita panggil adalah stop(1). Secara otomatis, kecepatan motor tersebut akan dijadikan nol, sehingga motor akan berhenti.

Selanjutnya adalah fungsi untuk menerima dan mencacah data serial :

```

data[0] = getchar();
if(data[0] == '#')
{
    for(i=0; i<3; i++)
        data[i] = getchar();
}

```

Data serial ditangkap dengan fungsi getchar() dan disimpan pada variabel data[]. Variabel data[] merupakan variabel *array* yang menyimpan beberapa karakter. Karakter pertama adalah karakter protokol. Sedangkan karakter berikutnya adalah karakter yang memberikan informasi kepada *controller* tentang apa yang harus dilakukan.

Setelah menerima data serial dan mencacahnya menjadi beberapa karakter, selanjutnya akan dilakukan pengambilan keputusan dengan pengondisian. Berikut adalah kode yang terkait :



```

if (data[0] == 'h')
{
    stop(1);
    stop(2);
    lcd_gotoxy(0,1);
    lcd_putsf("....");
    delay_ms(1000);
    lcd_gotoxy(0,1);
    lcd_putsf(" ");
}

if (data[0] == 's')
{
    if(flagX == '1' || flagY == '1')
    {
        data[0] = flagDir;
        cepatX = 7;
        cepatY = 7;
    }
    else
    {
        stop(1);
        stop(2);
    }
    lcd_gotoxy(0,1);
    lcd_putsf(" ");
    lcd_gotoxy(0,1);
    lcd_putsf(" Locked");
}

if(data[0] == 'q')
{
    run(1,1,cepatX);
    run(2,1,cepatY);
    lcd_gotoxy(0,1);
    lcd_putsf(" ");
    lcd_gotoxy(0,1);
    lcd_putsf(" Kiri -- Atas");
}

```

Dari kode diatas, kita dapat mengetahui bahwa yang membawa informasi tentang aksi apa yang harus dilakukan adalah variabel data[0]. Maka, hasil pengecekan dari variabel tersebut akan menentukan aksi berikutnya yang harus dilakukan. Sebagai contoh, bila data[0] = 'q' , maka motor 1 akan digerakkan ke arah kiri dan motor 2 digerakkan ke arah atas. Begitu seterusnya.

### 3.2 Pengujian dan Analisa

Pada bab ini, akan dipaparkan beberapa hasil dari percobaan atau uji coba dari bagian-bagian penyusun sistem berikut hasil analisisnya. Selain itu, juga akan dipaparkan hasil uji coba dan analisa dari sistem penyusun robot secara keseluruhan.

Uji coba tersebut dibagi menjadi dua bagian yaitu bagian perangkat keras dan bagian perangkat lunak. Berikut ini adalah penjelasan dan pemaparan yang lebih mendetail dari sistem-sistem yang ada.

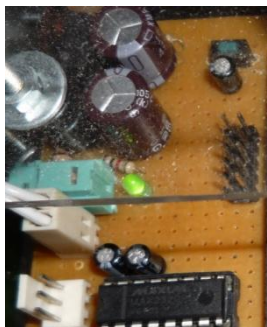
#### 3.2.1 Pengujian dan Analisa Perangkat Keras

Pada sub-bagian ini akan dipaparkan dan dianalisa hasil dari uji coba terhadap perangkat keras yang digunakan.

##### 3.2.1.1 Pengujian Mikrokontroler Atmega8535

Uji coba *Minimum System* ini dapat dilakukan dengan memasukkan program-program sederhana seperti menyalakan LED dan lain-lain.

Untuk rangkaian Supply, pengujian dapat dilakukan dengan dipasangnya LED sebagai indikator adanya arus atau tidak. Pemasangan LED ini disengaja agar menyala jika saklar ditekan.



**Gambar 3.11** Led menyala

Pada gambar diatas, dapat kita lihat sebuah LED yang menyala, dikarenakan saklar ditekan, dan sebaliknya, akan mati bila dalam kondisi off.

### 3.2.1.2 Pengujian dan Analisa Modul LCD

Uji coba kedua dilakukan pada modul LCD. Fungsi dari penggunaan modul LCD ini adalah untuk menampilkan tulisan-tulisan yang bisa berupa informasi penting ataupun status-status yang ada.

Cara pengujian terhadap modul LCD ini adalah dengan cara mencoba menampilkan tulisan pada modul LCD. Pertama kali yang dilakukan adalah memasukkan program pengujian modul LCD ke Atmega 8535.

Program untuk menampilkan tulisan pada LCD adalah dengan meng-include-kan `lcd.h` pada *Codevision AVR*. Berikut ini adalah kode inti dari program pengujian modul LCD.

```
#include <mega8535.h>

//Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>

lcd_putsf("  -[EyeBot]-  ");
```

Maksud dari kode diatas adalah menggerakkan kursor ke posisi 0,0 dan menulis kalimat “ Gunnerbot Sukses ” pada LCD mulai dari posisi tersebut.

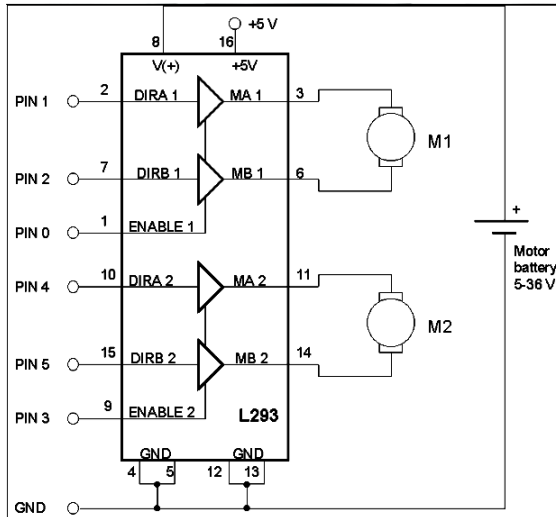
Bila rangkaian modul LCD belum benar, tidak akan muncul tulisan apa-apa. Sedangkan, bila rangkaian modul benar, maka akan muncul hasil seperti berikut ini.



**Gambar 3.12** Berhasil menampilkan tulisan pada LCD.

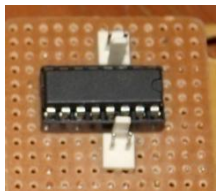
### 3.2.1.3 Pengujian dan Analisa Rangkaian Driver Motor

Berikutnya adalah pengujian pada rangkaian driver motor. Pengujian ini perlu dilakukan untuk mengetahui apakah rangkaian driver ini sudah dirakit dan berfungsi dengan baik.



**Gambar 3.13** Schematic driver motor

Rangkaian ini terdiri dari sebuah IC driver tipe L293D. IC ini memang didedikasikan sebagai IC driver motor. Untuk menguji rangkaian ini adalah dengan cara memberikan sinyal *High* ke salah satu kaki DIRA atau DIRB dan pada kaki ENABLE.



**Gambar 3.14** Komponen IC driver motor pada rangkaian

Pada percobaan ini, bila berhasil maka motor akan bergerak ke suatu arah tertentu. Berikut ini adalah tabel kebenaran dari rangkaian ini baik untuk driver 1 maupun 2.

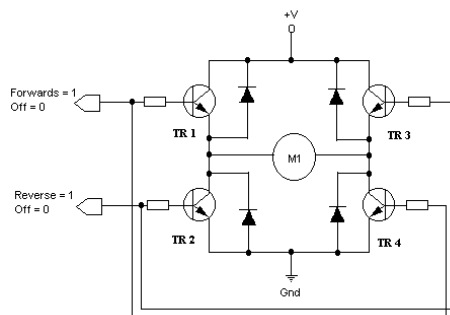
**Tabel 3.2** Tabel kebenaran arah gerak motor

Enable	DIRA	DIRB	Arah Putar
1	1	0	<i>Clock Wise</i>
1	0	1	<i>Counter clock wise</i>
1	0	0	Diam
1	1	1	<i>Break</i>

Pada tabel diatas yang ditampilkan hanya saat *enable* bernilai 1. Karena bila *enable* berlogika 0, maka apapun nilai dari kaki lain , motor akan tetap diam. Sedangkan apabila pada saat *enable* berlogika 1 dan DIRA serta DIRB juga berlogika 1, maka akan terjadi pengereman. Sedangkan bila pengereman berlangsung cukup lama, maka IC L293D akan menjadi panas.

### Analisa

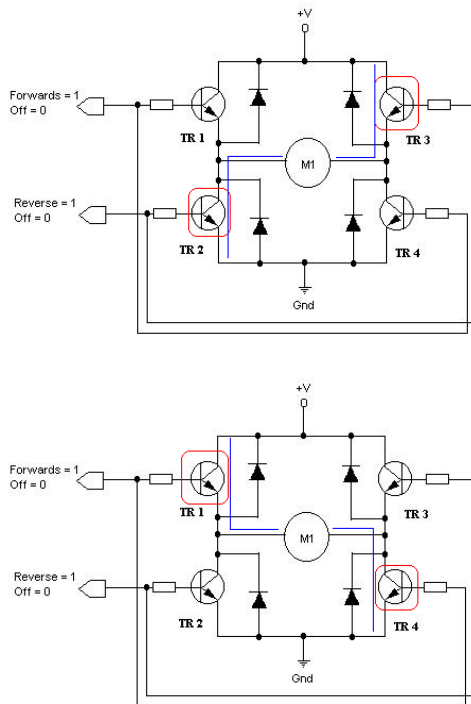
Berikut ini adalah analisa dari rangkaian driver yang digunakan. Pada prinsipnya, rangkaian driver ini bekerja layaknya sebuah H-Bridge yang terdiri dari 4 buah transistor NPN dibawah ini. Fungsi dari transistor NPN pada kasus ini adalah sebagai saklar elektronik.



**Gambar 3.15** Ilustrasi driver motor

Untuk mengaktifkan sebuah transistor, yang perlu dilakukan adalah memberi tegangan pada kaki basis sebesar lebih dari 0,6 volt. 0,6 volt sendiri adalah tegangan penghalang yang dimiliki oleh kaki basis.

Setelah transistor aktif, maka kaki emiter dan kolektor akan seolah-olah terhubung layaknya sebuah saklar yang ditekan. Pada kasus ini, saklar yang disusun dengan susunan seperti rangkaian diatas akan mengalirkan arus ke motor DC dalam arah yang berbeda sehingga mengakibatkan motor berputar ke arah yang berbeda pula.



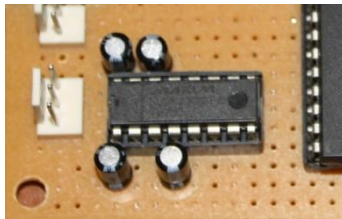
**Gambar 3.16** Transistor yang aktif dan arah arus listrik

Pada gambar diatas, dapat kita lihat cara kerja driver secara prinsip. Transistor yang diberi tanda merah adalah transistor yang aktif. Sedangkan garis warna biru merupakan arah

arus yang terjadi dari positif ke negatif. Dapat kita lihat arah arus yang masuk dan keluar dari motor berbeda antara gambar atas dan bawah. Hal tersebut menyebabkan arah putaran motor yang berbeda pula.

#### 3.2.1.4 Pengujian dan Analisa Sistem Komunikasi Serial

Pengujian terakhir dilakukan pada rangkaian transmisi serial. Pada dasarnya, rangkaian ini terdiri dari sebuah IC serial. IC ini bertugas meneruskan data serial yang masuk.



**Gambar 3.17** IC serial pada rangkaian.

Pengujian pada rangkaian ini dilakukan dengan cara mencoba mengirim data serial dari PC dan menampilkannya pada modul LCD. Berikut ini adalah program yang menjalankan tugas tersebut.

```

lcd_init(16);
lcd_gotoxy(4,0);
lcd_putsf("--[EyeBot]--");
while (1)
{
    for(i=0;i<16;i++)
    {
        data[i]=getchar();
    }
    lcd_gotoxy(7,1);
    lcd_puts(data);
}

```

Bila pengujian ini berhasil, maka akan muncul tulisan pada modul LCD. Berikut adalah gambar yang menunjukkan tampilan dari LCD.



**Gambar 3.18** Menampilkan tulisan melalui serial

### **Analisa**

Analisa pada percobaan ini dilakukan pada IC serial. IC serial ini akan selalu meneruskan dan memperkuat sinyal data serial yang melewatinya. Dengan demikian, data akan sampai dengan utuh dan dapat dimengerti oleh pasangan komunikasi.

Pengiriman data ini dilakukan per-bit. Setelah semua bit diterima, data ini akan dirakit kembali dan dibaca.

### **3.2.2 Analisa Perangkat Lunak**

Setelah pengujian perangkat keras, pada sub-bagian ini akan dipaparkan tentang analisa perangkat lunak dari sistem. Pengujian ini dibagi menjadi 2 bagian lagi. Yaitu bagian *PC side* atau perangkat lunak yang berjalan di PC, dan *Controller Side* atau program yang berjalan di *microcontroller*.

#### **3.2.2.1 Analisa Program pada PC**

Pada bagian ini, akan ditunjukkan hasil analisa dari program di sisi PC. Program yang dimaksud ada dua bagian :

1. Bagian komunikasi serial.
2. Bagian Deteksi wajah.

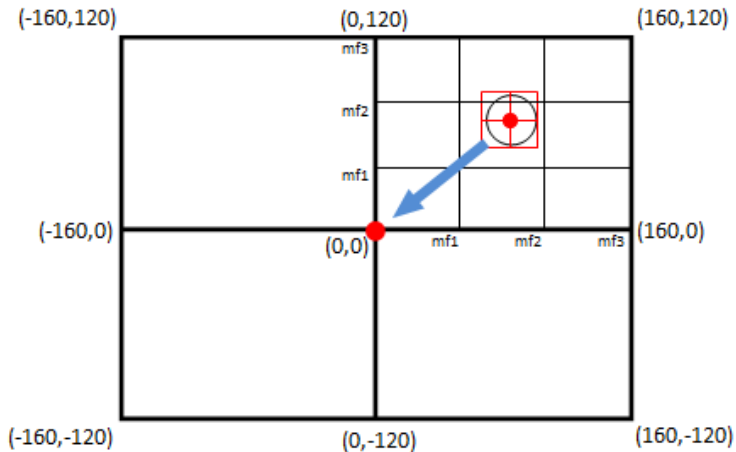
Bagian pertama bertugas dan bertanggung jawab pada bagian transmisi data. Bagian ini bertugas mulai inisialisasi *serial port*, menyiapkan data serial, dan mengirim data serial.

Pada proses mempersiapkan data, program ini sangat bergantung pada perhitungan di program deteksi wajah. Hal itu dikarenakan data yang akan dikirim tersebut adalah hasil dari pengambilan keputusan pada bagian deteksi wajah.

Program deteksi wajah tersebut berupa aplikasi konsol, di dalamnya terdapat beberapa perhitungan mengenai titik pusat objek dan titik tengah frame kamera yang nantinya akan menentukan pada proses pengiriman data ke bagian mikro.



Berikut ini adalah simulasi tampilan frame sebelumnya,



**Gambar 3.19** Simulasi tampilan frame

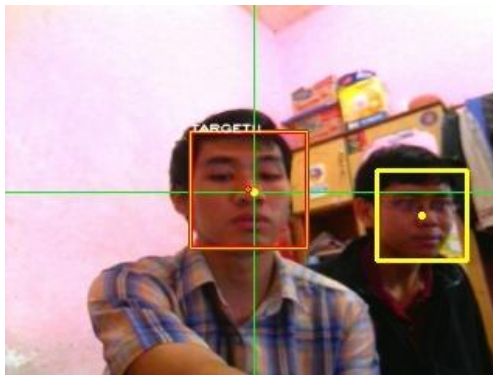
Dari gambar di atas, terlihat frame kamera dibagi dalam 4 kuadran dengan titik  $(0,0)$  dimisalkan sebagai pusat frame. Misalkan ada objek yang berada pada kuadran I dengan titik  $(160 \times 120)$  sebagai titik terjauh, program akan menentukan kecepatan motor dalam mengejar objek, dimana semakin jauh pusat objek dari pusat frame, maka semakin cepat motor bergerak, namun jika dekat, motor akan bergerak lambat untuk menyamakan pusat frame dengan titik pusat objek.

Berikut ini adalah *screen shoot* pengujian program deteksi wajah :



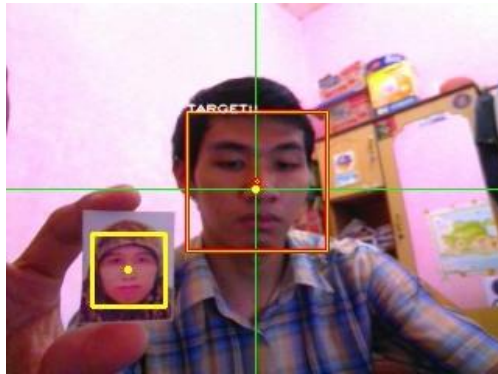
**Gambar 3.20** Pengujian 1

Dari hasil pengujian 1 di atas, program berhasil mendeteksi objek single dan langsung menjadikannya sebagai target.



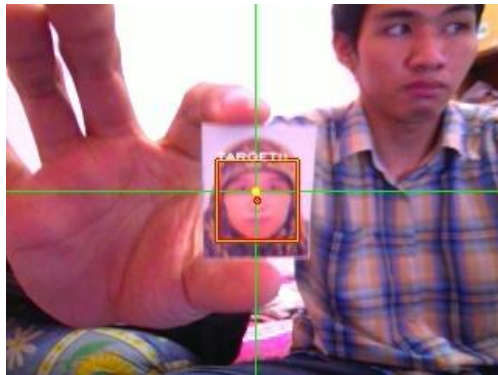
**Gambar 3.21** Pengujian 2

Dari hasil pengujian 2 di atas, Program berhasil mendeteksi banyak objek dan memprioritaskan satu objek untuk dijadikan sebagai target



**Gambar 3.22** Pengujian 3

Berikut ini adalah salah satu kelemahan dari program deteksi wajah, karena seluruh wajah yang ada pada frame akan dideteksi baik itu adalah wajah manusia atau hanya sebuah foto wajah.



**Gambar 3.23** Pengujian 4

Kelemahan yang lain adalah objek wajah yang ada pada tiap frame harus utuh, dan tidak boleh terpotong, miring, atau terhalang oleh sesuatu.

Berikut ini adalah screen shoot dari aplikasi konsol yang menjelaskan kondisi objek yang terdeteksi, koordinat pusat objek pada frame, dan komunikasi serial yang terjadi.

```

D:\DataQ\Lain2\Documents\Visual Studio 2008\Projects\EYEBOT\Debug\EyeBot.exe --casca...
Sumbu X : +      Sumbu Y : +      Pusat <193,111>      Sending : I e 2 2 1
Sumbu X : +      Sumbu Y : +      Pusat <193,111>      Sending : I e 2 2 1
Sumbu X : +      Sumbu Y : +      Pusat <192,112>      Sending : I e 2 2 1
Sumbu X : +      Sumbu Y : o      Pusat <193,114>      Sending : I d 2 2 1
No Target!!
No Target!!
No Target!!
No Target!!
No Target!!
Sumbu X : +      Sumbu Y : -      Pusat <263,130>      Sending : I c 1 2 1
No Target!!

```

**Gambar 3.24** Aplikasi konsol

Dari gambar 3.24 di atas, terdapat tulisan Sumbu X : + dan Sumbu Y = +, keduanya itu menjelaskan posisi dari pusat target terhadap sumbu x dan y. Ada kalanya keduanya menunjukkan titik 0 dan 0, itu berarti antara pusat target dengan pusat frame sudah berimpit. Untuk tulisan Pusat (193,111) itu menunjukkan koordinat pusat target pada frame, dimana pusat frame sebagai acuan titik (0,0). Tulisan sending menunjukkan kondisi pengiriman karakter ke program mikro.

### 3.2.2.2 Analisa Program pada kontroler

Perangkat lunak bagian kedua adalah perangkat lunak yang bekerja pada rangkaian *microcontroller*. Perangkat lunak atau program ini bertugas menerima perintah dari PC dan menjalankan perintah tersebut.

Disini, terdapat beberapa bagian fungsi dari program. Beberapa bagian tersebut diantaranya:

1. Membaca data serial.
2. Mengambil keputusan berdasarkan data serial yang diterima.
3. Mengendalikan peralatan luar.

Fungsi pertama adalah membaca data serial. Berikut ini adalah contoh data serial yang dikirim.

```
data_serial = {'#', 'e', '2', '1'};
```

Karakter pertama dari data tersebut berfungsi sebagai penanda bahwa 3 karakter berikutnya adalah data penting.

Kemudian, karakter kedua merupakan kode untuk arah gerak robot. Dan dua data berikutnya merupakan data untuk menentukan kecepatan motor 1 dan motor 2.

Sedangkan kode untuk membaca data serial tersebut adalah sebagai berikut.

```
data[0] = getchar();
if(data[0] == '#'){
    for(i=0;i<3;i++)
        data[i] = getchar();
}
```

Kode pada baris pertama berfungsi untuk mendeteksi adanya karakter penanda. Bila karakter tersebut diterima, maka 3 karakter berikutnya akan dibaca.

Setelah semua data diterima, kemudian akan dilanjutkan dengan pemilihan keputusan yang tepat berdasarkan data tersebut. Kode untuk memilih keputusan tersebut adalah sebagai berikut.

```
if(data[0] == 'q'){
    run(1,1,cepatX);
    run(2,0,cepatY);
    lcd_gotoxy(0,1);
    lcd_putsf("                ");
    lcd_gotoxy(0,1);
    lcd_putsf("Kiri - Atas");
}
if (data[0] == 'w'){
    run(2,0,cepatY);
    stop(1);
    lcd_gotoxy(0,1);
    lcd_putsf("                ");
    lcd_gotoxy(0,1);
    lcd_putsf("Atas");
}
```

Pemilihan keputusan ini dilakukan dengan menggunakan perintah *if* dan *else*. Yang diperiksa pada proses ini adalah deret data ke 2 dari deretan data yang ada.

Langkah berikutnya dalam mengendalikan peralatan luar. Kode untuk mengendalikan peralatan luar juga nampak pada kode diatas. Kode yang dimaksud terletak pada baris setelah kontrol *if*.

Pada pemanggilan fungsi `run(1,1,cepatX)` , yang akan terjadi adalah motor 1 akan digerakkan kearah kiri dengan kecepatan sebesar nilai variabel `cepatX`. Sedangkan perintah `lcd_putsf()`, bertujuan untuk menampilkan tulisan di LCD. Dan perintah `stop()`, berfungsi untuk menghentikan putaran motor. Langkah-langkah tersebut dilakukan berulang-ulang atau *loop* hingga robot dimatikan.

### 3.2.3 Uji coba Sistem Secara Keseluruhan

Bagian ini akan menyajikan beberapa pengujian sistem secara keseluruhan. Sistem secara keseluruhan sendiri meliputi *Program pada PC* dan program pada kontroler serta interaksi diantaranya.

**Tabel 3.3** Pengujian Deteksi wajah berdasar jarak

No.	Jarak Target (cm)	Berhasil ?
1	10	Berhasil
2	20	Berhasil
3	30	Berhasil
4	40	Berhasil
5	50	Berhasil
6	60	Berhasil
7	70	Berhasil
8	80	Berhasil
9	90	Berhasil
10	100	Berhasil
11	110	Gagal
12	120	Gagal
13	130	Gagal
14	...	Gagal

Pengujian di atas, dimaksudkan untuk mengetahui seberapa jauh program deteksi wajah dapat bekerja dengan baik. Program ini akan berhasil selama objek wajah utuh tanpa halangan berada dalam lingkungan frame kamera dengan jarak tertentu.

Dari data di atas dapat diketahui bahwa program masih dapat mendeteksi adanya objek wajah dalam interval jarak kurang dari 1 meter saja. Hal ini dipengaruhi oleh faktor cahaya pada saat pengujian dilakukan. Kamera web pada umumnya sangat dipengaruhi cahaya untuk hasil yang di dapatkan, untuk itu sebaiknya pemilihan kamera yang memiliki kepekaan tinggi terhadap cahaya sangat dianjurkan agar sistem dapat bekerja lebih baik lagi.

Pengujian berikutnya adalah untuk mengetahui seberapa cepat respon sistem perangkat keras dalam mengikuti objek yang dijadikan sebagai target. Pengujian kali ini dilakukan pada interval jarak objek berhasil terdeteksi dengan baik, namun berbeda tempat tempat kuadran dalam frame kamera.

**Tabel 3.4** Pengujian kecepatan respon sistem

<b>No.</b>	<b>Posisi Target pada frame</b>	<b>Jarak Thd center</b>	<b>Respon</b>
1	Kanan , Atas	Dekat	Lambat
2	Kanan , Atas	Jauh	Cepat
3	Atas	Dekat	Lambat
4	Atas	Jauh	Cepat
5	Kiri , Atas	Dekat	Lambat
6	Kiri , Atas	Jauh	Cepat
7	Kiri	Dekat	Lambat
8	Kiri	Jauh	Cepat
9	Kiri , Bawah	Dekat	Lambat
10	Kiri , Bawah	Jauh	Cepat
11	Bawah	Dekat	Lambat
12	Bawah	Jauh	Cepat
13	Kanan , Bawah	Dekat	Lambat
14	Kanan , Bawah	Jauh	Cepat
15	Kanan	Dekat	Lambat
16	Kanan	Jauh	Cepat

Dari percobaan di atas, program berjalan sesuai dengan apa yang diinginkan. Pada saat target berada pada sisi terdekat dari pusat frame, maka motor akan bergerak lambat, hal itu dilakukan untuk meminimalisasi over posisi kamera dalam mengikuti objek. Pada saat objek berada pada posisi yang jauh dari pusat frame maka awalnya motor akan bergerak cepat, kemudian jika sudah mendekati objek yang dimaksud, kecepatan motor akan menurun.

Untuk mengatasi masalah respon dari motor, dan mengantisipasi over posisi dari motor, sebaiknya digunakan motor yang lebih presisi, biasanya dipakai dalam dunia robotika.



## **BAB IV**

## **PENUTUP**



## BAB IV

### PENUTUP

Pada bagian akan dibahas mengenai hasil dan kelemahan dari alat yang telah dibuat. Setelah melakukan perencanaan, pembuatan dan implementasi sistem EyeBot, kemudian dilakukan pengujian dan analisa maka dapat diambil kesimpulan dan saran- saran sebagai berikut:

#### 1. KESIMPULAN

Dari hasil uji coba sistem ini dapat ditarik beberapa kesimpulan:

- Cara kerja sistem dalam mengejar target adalah dengan membandingkan titik pusat target dengan titik pusat *frame* kamera, kemudian memperkecil *error* secara terus-menerus sampai didapatkan nilai yang sama.
- Program deteksi wajah yang telah dibuat memiliki banyak kelemahan dan harus diperbaiki.
- Motor yang digunakan kurang responsif sehingga pembidikan target sangat lambat dan kadang-kadang kurang tepat sasaran.
- Kamera yang digunakan kurang baik, sehingga mempengaruhi respon terhadap target yang tentunya menjadi lebih lambat.

#### 2. SARAN

- Program harus diperbaiki lagi, dengan cara membuat klasifikasi sendiri tentang objek yang akan dideteksi.
- Kamera yang digunakan sebaiknya kamera yang lebih bagus, yaitu kamera dengan resolusi 5Mp ke atas, atau memakai kamera CCTV, karena itu akan menentukan seberapa jauh objek dari robot tetap terdeteksi dan juga kamera harus memiliki respon yang lebih cepat dalam meng-capture gambar.
- Motor yang digunakan untuk eksperimen berikutnya sebaiknya motor yang standart dipakai dalam dunia robotika, sehingga gerakannya dapat lebih presisi.
- Bahan konstruksi sebaiknya dibuat dengan bahan dasar logam terutama untuk bagian-bagian yang merupakan sambungan poros dari motor, sehingga robot dapat lebih stabil.



## DAFTAR PUSTAKA

- [1] “openCVsources”, (<http://sourceforge.net/projects/opencvlibrary>.)
- [2] “openCV”, (<http://osc.ugm.ac.id/site/index.php/2008/06/opencv/>.)
- [3] “steppermotor”, (<http://blog-elektronika/2009/>)
- [4] “project”, ( <http://vasc.ri.cmu.edu/NNFaceDetector/>)
- [5] “finalproject”, ([http://www.cc.gatech.edu/~kihwan23/imageCV/Final2005/FinalProject\\_KH.htm](http://www.cc.gatech.edu/~kihwan23/imageCV/Final2005/FinalProject_KH.htm))
- [6] Muhammad Sadeli, 2009, “Visual Basic.net 2008”, Maxikom
- [7] Abdul Kadir, (2004) Pemrograman visual C++ , ANDI Yogyakarta.
- [8] Rusmadi, Dedi. 2006. Teknik Interfacing Port Serial dan Paralel. Yogyakarta: Graha Ilmu.
- [9] Ahmad, Usman. 2005. Pengolahan Citra Digital dan Teknik Pemrogramannya. Yogyakarta: Graha Ilmu



